PROJECTE FINAL DE CARRERA

INTERNAL DELAY CALIBRATION OF ERS-1/2 RADAR ALTIMETERS

(CALIBRATGE DEL RETARD INTERN DELS ALTÍMETRES D'ERS-1/2)

Estudis: Enginyeria de Telecomunicació

Autor: Roger Escolà Jané

Directors: Mònica Roca Aparici

Antoni Broquetas Ibars

Any: 2016

Table of Contents

Collaborations

This project was entirely carried out at isardSAT, during my internship of over 9 months in 2010 and 2011.

Acknowledgements

This project would not have been possible without *Mònica* and *Bernat*, who guided me and shared their knowledge wih me through the whole process.

I would also like to thank...

Pablo, for all the help he has offered since the first day we met;

Nando, for the memorable moments at the very beginning;

Lluís and *Mònica*, for giving me the opportunity to work in isardSAT;

Toni, the project co-director, for making things easier in a very tight schedule;

Carla, who has been a great support during all the long summer weekends at home;

Aven, for always showing us the light at the end of the tunnel;

Anna, my sister, for the mediation she has done; and

Alba, for having sacrificed a whole summer without complaining, for always being there and taking care of me.

And I would like to dedicate this project to *my parents*, for their unconditional support.

Resum del projecte

En les últimes dècades, s'ha dedicat molt d'esforç en intentar entendre el canvi climàtic, identificar-ne les causes i decidir com pot ser pal·liat. D'entre tots aquests esforços, l'Agència Espacial Europea (ESA) va crear el projecte Climate Change Initiative per tal de monitoritzar diverses variables climàtiques essencials (ECV). Una d'aquestes variables és el nivell del mar.

El nostre estudi ajudarà a millorar la precisió i la qualitat de l'estimació del nivell del mar proporcionant millors dades del calibratge intern dels altímetres de les missions ERS-1 i ERS-2. Això s'aconseguirà estudiant quin és el millor mètode per estimar el retard de la resposta impulsional de l'altímetre i, posteriorment, reprocessant els 12 anys de dades altimètriques disponibles.

Abstract

For the last few decades, a lot of effort has been put in trying to understand how the Earth's climate is changing, identify its causes and determine how it can be mitigated. Among all these efforts, ESA created the Climate Change Initiative project in order to monitor several Essential Climate Variables. One of these variables is the Sea Level.

Our study will help improve the precision and quality of the Sea Level variable by providing better results on the altimeter's internal calibration for the entire ERS-1 and ERS-2 missions. This will be achieved by studying which is the best method to compute the internal path delay of the altimeters and, after that, reprocessing the 12 years of available altimetry data.

Chapter 1

Introduction

The term 'weather' is defined as an atmospheric condition at any given time or place [1]. It is measured in terms of wind, temperature, humidity, atmospheric pressure and precipitation. On the other hand, 'climate' is usually defined as the "average weather" [2], or more rigorously, as the statistical description in terms of the mean and variability of weather over a period of time ranging from months to thousands or millions of years[1]. Then, a significant change in these patterns, lasting for an extended period at either local or global scale, is what is called a 'climate change'.

This change may result from natural factors, such as sun's intensity or slow changes in the Earth's orbit around the sun; natural processes within the climate system (e.g. changes in ocean circulation); and human activites that change the atmosphere's composition and the land surface. However, the UNFCCC (United Nations Framework Convention on Climate Change), on its first article [4], restricts this concept to "a change of climate which is attributed directly or indirectly to human activity that alters the composition of the global atmosphere and which is in addition to natural climate variability observed over comparable time periods".

[1]Definition from the IPCC's Fourth Assessment Report: Climate Change 2007, [3]. The IPCC, Intergovernmental Panel on Climate Change, is the international leading body for the assessment of climate change, established by the United Nations Environmental Programme (UNEP) and the World Meterological Organization (WMO) in 1988.

During the last few decades, a big effort has been put to understand the climate change, identify its causes and determine how it can be mitigated. However, there are still many uncertainties that have not let the scientific community to reach a final conclusion.

1.1 State of the art

Since the adoption of the United Nations Framework Convention on Climate Change (UNFCCC) in 1992, the Global Climate Observing System (GCOS, also established in 1992), took the responsibility of providing a systematic observation and development of data archives related to the climate system. In this line of work, in 2003, the GCOS established a list of Essential Climate Variables (ECV) that were feasible and highly related to the UNFCCC requirements. In 2006, as a result from the dialogue with space agencies (as satellite observations provide unique information that help to successfully understand the climate change), a set of requirements for satellite-based products were defined and agreed globally. Later in 2008, ESA, the European Space Agency, in order to provide a consistent set of ECVs, initiated a new programme called Global Monitoring of Essential Climate Variables, also known as the ESA Climate Change Initiative (ESA CCI, [5]).

In a first phase of the programme, 11 ECVs were chosen and one project per each ECV was set up. The ECVs were: aerosol properties, cloud properties, fire, greenhouse gases, glaciers, land cover, ocean colour, ozone, sea ice, sea level and sea surface temperature. Later, in a second phase (end 2011 - beginning 2012), two more variables were added to the programme: ice sheets and soil mosture.

In order to create reliable, tested, validated data products, a critical step of merging data from different sensors and sources had to be performed since no single data set from any single Earth Observation (EO) mission or space agency was sufficient to constitute independently all data required for a complete ECV. This means that each ECV project was formed by several parties and each ECV final data set would be built from a multi-mission, long-term data base. In this line of

work, it was mandatory to gather information from all possible missions. However, some of these data had to be reprocessed as they came from different sensors or were obtained in a distant past. This was the case of the Sea level ECV.

1.2 Motivation

Providing a long-time series of data, led the Sea level ECV project to be focused, among other key objectives, on performing a progressive integration of the ESA missions ERS-1, ERS-2 and Envisat. As this integration was too costly in all terms, a parallel project was created to support CCI: REAPER (REprocessing of Altimeter Products for ERs)[6].

REAPER's goal is to reprocess ERS-1 and ERS-2 altimetry data with the best available algorithms, correction models and auxiliary data to achieve the highest accuracy in the geophysical products. Moreover, precise orbits across the 12 year span have to be regenerated to improve accuracy and continuity. Besides the reprocessing of the data, another key step is to create output products which are as similar as possible in character and format to Envisat altimetry products in order to ease the use of the 16 years of data across the three ESA missions.

Within all the reprocessing algorithms that have to be held, we will put our effort on the calibration chain, specifically on the internal path delay. This delay has a direct impact on the surface height determination. A more precise computation of this internal delay would provide a more accurate value for the height of the surface. The motivation is then to decide which method will be used to re-compute the ERS radar altimeter's internal delay correction.

1.3 Goal of the work

The main purpose of the carried out work is to study which method to estimate the internal delay is theoretically the best. Afterwards, we will create a testing software that implements all the methods that have been proposed and allows us to compare the outputs. The testing will be held in different stages in terms of

time. This means that the test will be carried out over one single record, firstly, then over a whole orbit and finally over a complete cycle of data. The final aim is to, through the analysis of the obtained results during the testing phase, extract conclusions to decide the method that will be finally implemented and used to reprocess ERS data by ESA. Besides, another final objective is to observe the direct impact of this reprocessing on the output products.

1.4 Overview of the report

This report is composed of five chapters.

- Chapter 1: Introduction (current)
- Chapter 2: Background. It contains an introduction to radar altimetry, giving information about its principle, its constraints and the way the data is processed as well as the corrections applied to it.
- Chapter 3: Internal delay calibration. This is the main chapter of the report which, in turn, is divided in three different parts:
  - The Point Target Response
  - PTR fitting methods
  - Testing
- Chapter 4: Conclusions. It provides a general analysis and some perspective on all the work that has been carried out.
- Chapter 5: Annex. It contains additional information on the work that has been performed within this report.

Chapter 2

Background

In general, altimetry is understood as the branch of topography that measures height (or altitude) with respect to a reference level. Depending on the science field of work this reference may change. For instance, in geology, most measurements of height are based upon the mean sea level[1] whereas in geodesy, height is preferred to be defined from the reference ellipsoid[2]. However, height and altitude may define different concepts. In aviation and spaceflight, height is measured from the overflown terrain and altitude can be referenced either to sea level or the reference ellipsoid.

Depending on the context, altitude can be measured with different methods and instruments. Hikers and climbers, for example, can use either a hand-held barometric altimeter, which computes the altitude by measuring the atmospheric pressure, or a GPS receiver, although the latter is less reliable and often less accurate. Aircrafts, in order to assist their navigation, use aneroid barometers, which are calibrated to show the pressure directly as the altitude above the mean sea level. On the other hand, as far as Earth observation is concerned, altitude is measured with radar[3]. Thus, the technique receives the name of radar altimetry.

[1]The mean sea level, MSL, is a measure of the average height of the ocean's surface over all the oceans.

[2]The reference ellipsoid is a mathematically-defined surface that approximates the Earth's shape. It is basically a sphere "flattened" at the poles.

[3]RADAR, RAdio Detecting And Ranging.

2.1 Radar altimetry

The purpose of radar altimetry is to determine the distance between the aircraft where the radar is mounted and a target surface. This is achieved by measuring the round trip time that takes the radar pulse to reach the surface and return to the receiver. However, the radar altimeter also measures the power and the shape of the reflected radar pulses, from which a lot of other information can be extracted.

2.1.1 Main objective

One of the main goals in radar altimetry is to determine the topography of the surface, namely, the surface height. The surface height is the distance between the reference ellipsoid and the actual target surface. Thus, it can be computed as the difference between the satellite's altitude (that has also been computed with respect to the reference ellipsoid) and the satellite-to-surface range.

The satellite-to-surface distance is inferred from the two-way delay of the radar echo from the Earth's surface. The radar transmits microwave pulses which propagate at a speed determined by the refractive index of the propagation medium, which is close to unity. The time elapsed from the transmission of a pulse to the reception of its echo reflected from the Earth's surface is proportional to this range, as shown in Equation 2.1.1:

$$t = \frac{2 \cdot r}{v} = \frac{2 \cdot r}{c/n} \tag{2.1.1}$$

where t is the time elapsed from transmission to reception, r is the satellite-to-surface range, c is the speed of light in vacuum and n is the refractive index of the propagation medium. Obviously, this time has to be corrected by atmospheric distortion and instrumental errors (see Section 2.1.5).

On the other hand, the satellite's altitude is the distance from the satellite to the reference ellipsoid. It has to be determined with the greatest possible accuracy in order to determine its precise orbit. Since the altitude depends on many external factors such as inclination, gravity forces, atmospheric drag, etc., the satellite has

to be tracked in many ways so as to meet the high precision requirements. The main tracking instruments are:

- DORIS[4] or PRARE[5] instruments, which accurately determine the satellite's velocity on its orbit, using dynamic orbitography models to deduce the satellite's trajectory relative to Earth.
- Star trackers, that calculate the satellite's attitude by matching triangle patterns of stars with patterns stored in their catalogue.
- Laser Retro-Reflectors (LRR), which are passive optical devices for ground-base measurement of the satellite orbit by means of Satellite Laser Ranging (SLR) stations.
- GPS, the known Global Positioning System.

After the satellite-to-surface range and the satellite's altitude have been properly computed, the surface height can be determined as in Equation 2.1.2:

$$SH_{corr} = alt - r_{corr} \tag{2.1.2}$$

where SH_{corr} is the corrected Surface Height, alt is the satellite's altitude and r_{corr} is the satellite-to-surface corrected range.

When the overflown surface is ocean, the surface height (or Sea Surface Height, SSH) integrates two important effects such as: the variation of the sea surface caused by mass distribution on Earth -the geoid- and the variation related to ocean dynamics (or dynamic topography). This last contribution comprises both a permanent stationary component (tides, permanent winds, etc.) and a highly variable component due to wind, eddies, seasonal variations, etc.

Besides the surface height, by analysing the reflected signal's amplitude and shape, the backscatter coefficient of the surface and its roughness can also be extracted. Moreover, when the target surface is an ocean, wave height and wind speed can be measured too.

[4]Doppler Orbitography and Radiopositioning Integrated by Satellite.
[5]Precise Range And Range-rate Equipment.

2.1.2 The principle

The radar altimeter is a pulse width limited radar. This means that the illumination of the target is limited by the radar pulse width, rather than the beamwidth. The radar signal is a narrow pulse which is transmitted in a narrow beam vertically down towards the surface. It is radiated in a spherical shell whose intersection with the surface defines an instantaneous illuminated area. When this thin shell of microwave radiation illuminates the flat surface of the ocean, the illuminated area is, at first, a rapidly expanding disc [7]. This is shown in Figure 2.1.1.

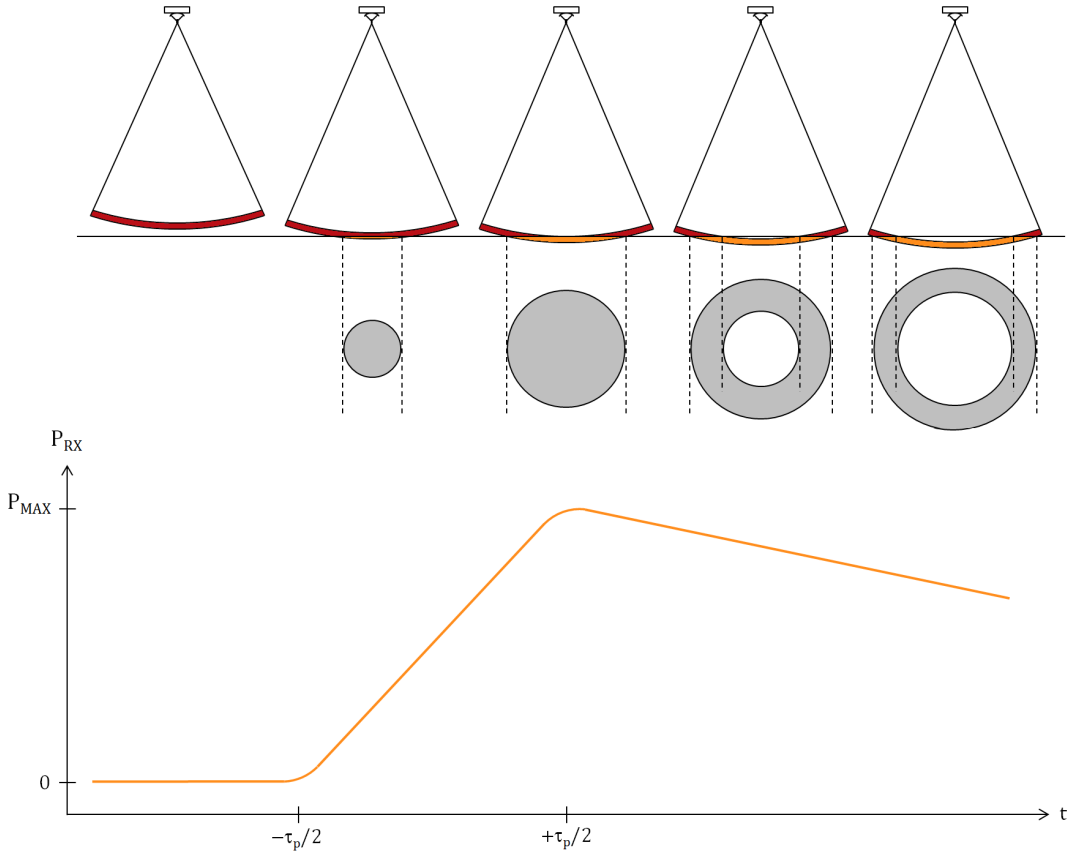


Figure 2.1.1: Simplified geometry of the return signal waveform. Between $-\tau_p/2$ and $+\tau_p/2$, the backscattered power rises. After that, the illuminated area remains constant but, because of the attenuation of the antenna pattern, the echo power decreases.

Soon the illuminated area becomes annular and from this point the illuminated

area (and thus echo power) remains constant. However, as the annulus expands, the incident and received scattered power become attenuated by the antenna pattern.

2.1.3 Range resolution

In radar technology[8], the capability of the pulse to discriminate neighbouring targets is called range resolution. Considering the sensing (uncompressed) pulse has a length of τ_u and that, for simple pulses, the bandwidth of a pulse is the inverse of its duration[6], the range resolution is defined as:

$$\Delta r = \frac{c \cdot \tau_u}{2} = \frac{c}{2 \cdot B} \tag{2.1.3}$$

Thus, if there were two point targets separated by a distance of Δr, the two received echoes would be separated in time by $\Delta t = 2 \cdot \Delta r / c$. Then, in order to be able to distinguish these two targets, τ_u has to be shorter than Δt. Otherwise, the two echoes would overlap and lead to the perception that there is only one dispersed target. This is depicted in Figure 2.1.2.

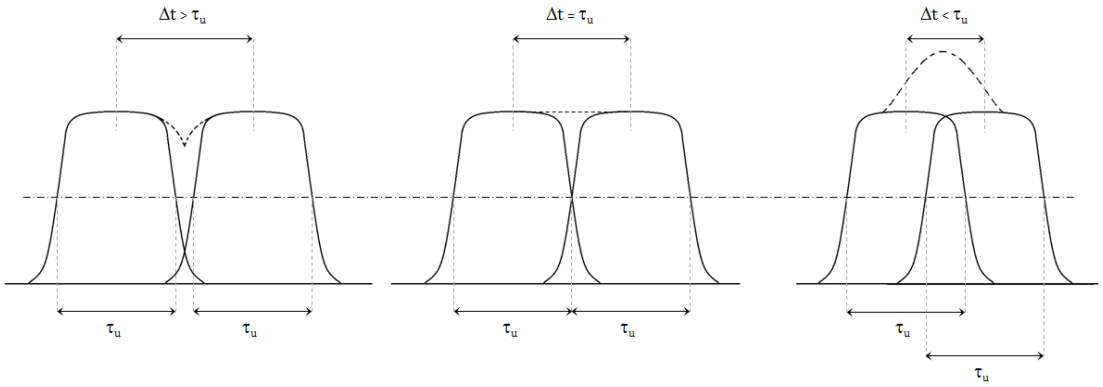


Figure 2.1.2: Range resolution concept. In the first case (left), the two targets that generatated the echoes can be distinguished. The second case (center) is the threshold case: if the targets are this close (or closer, as in the third case; right), the radar will not be able to distinguish them.

[6]However, Equation 2.1.3 is only valid for square pulses in radars without pulse compression. In that case, the equation should be rewritten as $\Delta r = c \cdot \tau_c / 2 = c/2B$, where τ_c is the duration of the compressed pulse.

Due to the large distance between the satellite and the surface, pulses with high energy are needed to detect a target. Hence, being the energy of a pulse described in Equation 2.1.4, high peak power pulses are needed.

$$E = P \cdot \tau_u \tag{2.1.4}$$

However, the maximum power is limited by the radar hardware, particularly in spaceborne sensors. The other possibility, then, is to increase τ_u but with the drawback of losing resolution because, as we can deduce from Equation 2.1.3, a long pulse corresponds to a narrow bandwidth and a resulting poor resolution. Thus, even it may seem a contradiction, a pulse with large τ_u and large B is needed in order to have a high detection capability (large energy) and high resolution (large bandwidth). This can be achieved by pulse compression together with linear frequency modulation.

The mathematical expression for a frequency modulated signal, that is, a chirp signal, is given by:

$$s(t) = \begin{cases} A \cdot cos\left(2\pi\left(f_0 + \frac{\Delta f}{2 \cdot \tau_u} \cdot t\right) t\right) & 0 \leq t \leq \tau_u \\ 0 & otherwise \end{cases} \tag{2.1.5}$$

being A the amplitude, f_0 the carrier frequency, Δf the frequency span and τ_u the duration of the uncompressed pulse. Hence, its instantaneous frequency is:

$$f\left(t\right) = \frac{1}{2\pi}\frac{d\phi}{dt} = \frac{d}{dt}\left[f_0 \cdot t + \frac{\Delta f}{2 \cdot \tau_u} \cdot t^2\right] = f_0 + \frac{\Delta f}{\tau_u} \cdot t \tag{2.1.6}$$

Intuitively, the bandwidth is determined by $B = |(f_0 + \Delta f) - f_0| = |\Delta f|$.

If a long chirp signal (and hence a high energy pulse), when received, is passed through a dispersive delay line (which delays low frequencies more than high frequencies) all the frequencies in the pulse merge almost simultaneously[7], having

[7]It is important that the filter is adapted (or matched) to the frequency pass. If it delays too little, the compression is not fully reached. If it delays too much, then the pulse is again dispersed, not compressed.

a very short pulse. However, the resulting pulse (called compressed pulse) still has the energy of the input pulse. This process is what is called range compression.

The length of the compressed pulse is determined by the bandwidth B:

$$\tau_c = \frac{1}{B} \tag{2.1.7}$$

In order to know how much the pulse has been compressed, the concept of a compression factor (also knows as time-bandwidth product, TBP) arises. It is defined as[9]:

$$\rho = \frac{\tau_u}{\tau_c} = \tau_u \cdot \Delta f \tag{2.1.8}$$

The resulting very high RF bandwidth, of the order of some hundreds of megahertz, entails difficulties both in receiver matched filter implementation and, even if such a filter would be available, in video signal handling. To circumvent these problems the Full Deramp technique is used in combination with linear FM pulse compression.

2.1.4 Full Deramp Technique

The Full Deramp concept consists in mapping time offset into frequency offset. As a result, targets at a different range will give echoes at different frequencies. A brief description of this technique follows.

As shown in Figure 2.1.3, the chirp signal is generated, up-converted to the transmit frequency and then transmitted (red path in Figure 2.1.3). When the echo from the surface is expected to return (through the orange path in Figure 2.1.3), the chirp generator is activated again. This time the chirp signal has a different path (the blue path in Figure 2.1.3): it is up-converted and routed to the first mixer in the receiver (the Deramp mixer), where it becomes the local oscillator (LO) signal, simplifying the signal processing stages, that will work with a much-reduced bandwidth. As both inputs have the same frequency rate, the output frequencies are constant tones. [7, 10]. Thus targets with a different range

give echoes at different frequencies. Therefore, the range discriminator at the receiver can be implemented with a bank of contiguous filters. This is depicted in Figure 2.1.4 and Figure 2.1.5.

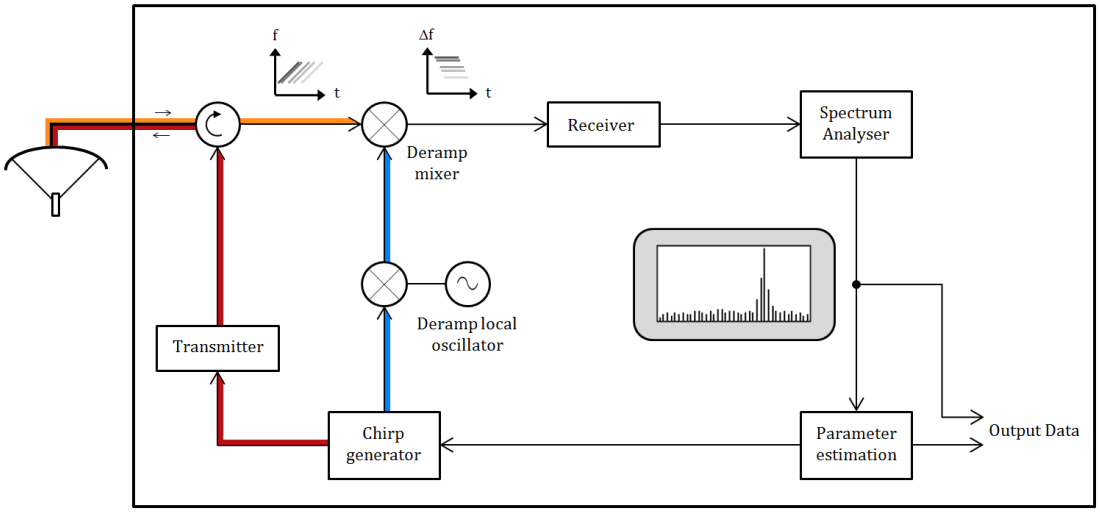


Figure 2.1.3: A simplified block diagram of the radar altimeter, showing the Full Deramp concept.

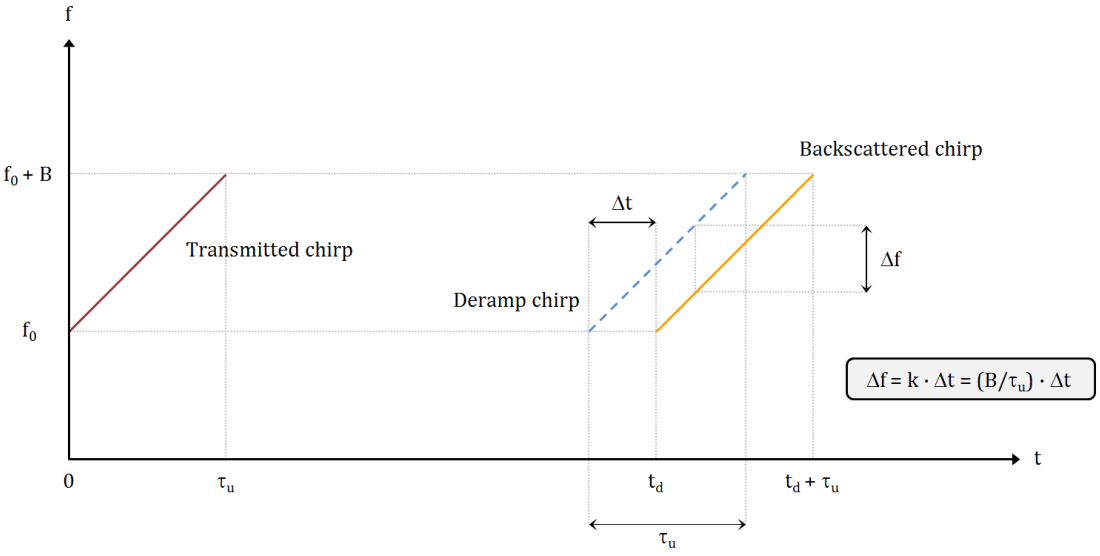


Figure 2.1.4: Full Deramp concept. The difference in time between the backscattered chirp and the Deramp chirp is translated to a difference in frequency.

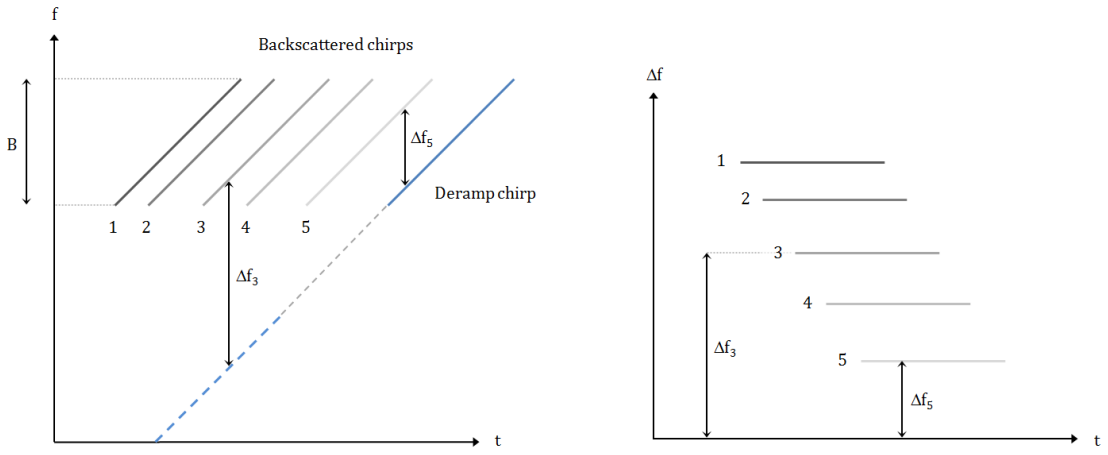


Figure 2.1.5: Conversion from time delays to constant frequency tones.

The Deramp technique can be explained if we consider the returns from two facets displaced by a distance Δh relative to each other. If the incident signal $s(t)$ is given by Equation 2.1.5 and its instantaneous frequency by Equation 2.1.6, the returned signals will have the following instantaneous frequencies:

$$f_1 = f_0 + \frac{B}{\tau_u} \cdot t \tag{2.1.9}$$

$$f_2 = f_0 + \frac{B}{\tau_u} \cdot (t + \Delta t) \tag{2.1.10}$$

where $\Delta t = 2 \cdot \Delta h / c$.

Then, as the received echoes are mixed with a chirp signal identical to the transmitted one, the resulting difference frequencies will be:

$$f_1^{'} = B \cdot \frac{t^{'}}{\tau_u} \tag{2.1.11}$$

$$f_2^{'} = B \cdot \frac{(t^{'} + \Delta t)}{\tau_u} = f_1^{'} + B \cdot \frac{\Delta t}{\tau_u} \tag{2.1.12}$$

Thus, the time delay Δt translates into a frequency shift Δf equal to

$$\Delta f = B \cdot \frac{\Delta t}{\tau_u} = B \cdot \frac{2 \cdot \Delta h}{\tau_u \cdot c} \tag{2.1.13}$$

The bandwidth has obviously to be dimensioned according to the maximum expected height range, Δh_{max}. Therefore, the IF bandwidth is given by

$$B_{IF} = \frac{B}{\tau_u} \cdot \Delta t_{max} \tag{2.1.14}$$

where $\Delta t_{max} = 2\Delta h_{max}/c$ is the gate duration for the maximum expected height range h_{max} .

2.1.5 On-board operations

The radar altimeter takes measurements from the surface. That occurs when it is working in tracking mode. However, in order to correctly take these measurements, it has to perform other tasks such as acquisition or in-flight calibrations.

2.1.5.1 On-board tracking and acquisition

So as to keep the waveforms well centred in range and power in the analysis window and to better adjust these parameters for the following echoes, the radar altimeter makes a brief processing of the recently recorded echoes. It anticipates the settings for the next echoes from the n previous ones. When this on-board tracking function is not able to adjust these parameters under critical conditions, the altimeter loses lock. Then, it enters in acquisition phase. During this phase, the radar altimeter searches for the signal, it locks onto it and stabilizes the tracking loops. During the acquisition sequence, there is no data until the tracking is properly reinitialised.

However, this procedure has been improved over the years. For instance, the Poseidon-3 altimeter (mounted on the NASA's satellite Jason-2) is equipped with an open-loop tracking technique for which a Digital Elevation Model (DEM) has been developed. The altimeter's onboard memory contains the elevation values of areas overlaid by the ground tracks. These data, combined with DORIS, are used to position the radar echo receiving window in advance, in order to anticipate the contrasts of the topography and to give priority to measurements over

water. This technique prevents the altimeter from losing track as sometimes occurs with a conventional tracking loop, and enables measurements to be acquired close to the shoreline or on continental waters. For the RA-2 altimeter (on ESA's Envisat), an innovative tracking algorithm, known as Model Free Tracker (MFT) [11] concentrates its effort in maintaining the earliest part of radar echoes within the tracking window, independently of its shape. In particular, the MFT will decide whether the range window is using the adequate resolution, whether the resolution could be increased or decreased, based on the Signal to Noise Ratio (SNR) of the on-board waveform position compared to reference values stored in the on-board memory. When the radar echo is about to move out of the tracking window, due for example to a sudden change in the surface elevation, the window is broadened to recapture the echo. This allows uninterrupted radar operation over all type of surfaces and its boundaries.

2.1.5.2 Calibrations

The instrument measurements are affected by many factors. These can be caused by instrument ageing effects as well as by thermal variations around the orbit and atmospheric distortion. These effects have to be measured and compensated for in the data. In order to be able to correct the affected measurements, the instrument performs a number of internal calibration procedures. These provide the following corrections:

- The Point Target Response (PTR) correction. This correction computes the instrument time delay and gain variations that have occured either in-flight after the preflight calibration or resulting from aging or changes in the onboard temperature. This correction will be further studied in Chapter 3.
- The Intermediate Frequency (IF) filter correction. The real instrument transfer function, which is ideally constant and equal to unity, needs to be measured by the instrument. This transfer function is mostly determined by the IF filter. This IF filter is measured by collection of noise spectra. This spectra is then inverted to generate a so-called *IF mask* that is used to correct the echo shapes for thermal effects.

- The UltraStable Oscillator (USO) correction. The USO clock is the most accurate time source on-board. With it, time measurements such as the PTR correction or the altitude measurements themselves can be compared. Although this oscillator, particularly in its temperature control, has high frequency stability, it also has a drift during the mission lifetime and it must be monitored and corrected. In order to perform this correction, the USO period is periodically downlinked to the ground where it is compared to an atomic frequency standard, that is, an atomic clock.

2.2 The ERS missions

The European Remote Sensing satellite ERS-1 was launched in 1991. Its main payload were an imaging synthetic aperture radar, a radar altimeter and other powerful instruments to measure ocean surface temperature and winds at sea. ERS-2, which overlapped with ERS-1, was launched in 1995 with an additional sensor for atmospheric ozone research [12].

At their time of launch the two ERS satellites were the most sophisticated Earth observation spacecraft ever developed and launched by Europe. These highly successful ESA satellites collected a wealth of valuable data on Earth's land surfaces, oceans, and polar caps and were called upon to monitor natural disasters such as severe flooding or earthquakes in remote parts of the world.

Shortly after the launch of ERS-2 in 1995 ESA decided to link the two satellites in the first 'tandem' mission which lasted for nine months. During this time the increased frequency and level of data available to scientists offered a unique opportunity to observe changes over a very short space of time, as both satellites orbited Earth only 24 hours apart.

In March 2000, a computer and gyro control failure led to ERS-1 finally ending its operations, after far exceeding its planned lifetime.

In July 2011, ERS-2 was retired and the process of deorbiting the satellite began.

Chapter 3

Internal delay calibration

Before being used by the scientific community, the science data received from the altimeter has to be corrected by several causes. Some of them are instrumental corrections, for example the USO (Ultra-Stable Oscillator), the IF-Mask and the PTR corrections, as mentioned in Section 2.1.5. These are made by some algorithms on the L1B chain[1], based on information collected on-board.

In this report we will focus on the PTR correction. Along ERS-1 and ERS-2, several methods to estimate the PTR have been proposed. Before implementing and testing which method is the best, and thus chosen for the final implementation, we will perform a graphical study in order to understand the whole calibration process. Therefore this main chapter is organised in three sections: a brief description of the PTR and the calibration chain, the presentation of the proposed fitting methods and finally the tests that we will perform in order to come to a final decision.

3.1 The Point Target Response

One of the on-board calibration data used on L1B is the Point Target Response (PTR) data. This data give information about the internal delay related to the

[1]The L1B chain is the first step in the processing of the raw satellite data. It produces an output that contains the same information as the satellite raw data, but geolocated and with instrument and calibration corrections applied.

travel of the chirp across the calibration path (the red path in Figure 3.1.1). This delay is later subtracted from the global round trip time that takes the signal to reach the surface and return to the receiver in order to not to take it into account when this time is converted to distance. Otherwise, as the round trip time would be greater than the real one, the calculated distance would be greater too and, hence, the obtained height would be lower. Thus, in order to ensure the highest accuracy in height it is very important to determine the internal delay as precise as possible.

In order to determine which waveform will be fitted, it is necessary to interpret the functional block diagram of the internal calibration chain (see Figure 3.1.1) and identify the points where the initial chirp is significantly modified. This calibration chain is also known, in ERS, as the Open-Loop Calibration (OLC) chain.

In the OLC chain, the transmitted chirp is delayed and passed via the Front-end electronics to the Deramp mixer, where the signal is mixed with the second chirp signal from the chirp generator, triggered by the parameter estimation block.

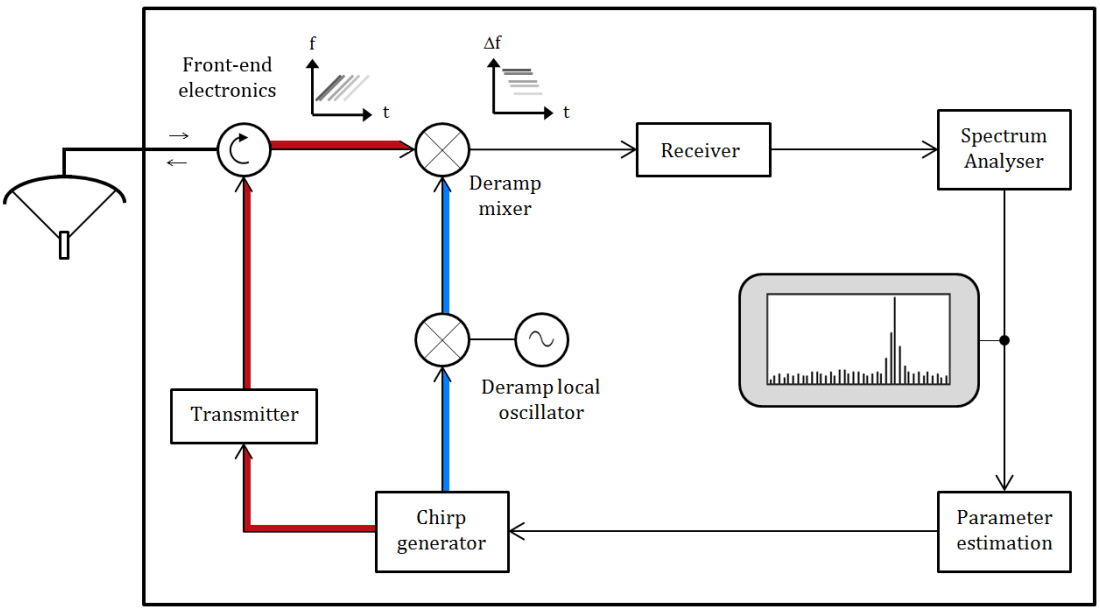


Figure 3.1.1: Steps followed by the original chirp in the PTR chain.

The resulting signal is a constant-frequency tone which is then modified by the receiver, where a signal weighting is applied to account for the finite window width: a Hamming window is applied, as defined in Equation (3.1.1), where $\alpha = 0.54$.

$$W_H(n) = \begin{cases} \alpha - (1-\alpha)cos\left(\frac{2\pi n}{N}\right) & 0 \leq n \leq N-1 \\ 0 & elsewhere \end{cases} \tag{3.1.1}$$

After that, the spectrum analyser performes a Fast Fourier Transform and some tracking parameters are estimated, which are then input to the chirp generator.

Through the path described above, the original chirp signal is modified. First, the transmitted chirp is multiplied by an identical chirp in the time domain, which means that both are convoluted in the frequency domain. The result of this convolution is a sinc-like waveform and is shown in Figure 3.1.2-1 (in FFT power units; hereafter expressed as FFT p.u.) and in Figure 3.1.2-2 (in dB). Then this signal is multipied by the Hamming window (Figure 3.1.2-3), again in the time domain; so convoluted in the frequency domain. This signal weighting reduces the sidelobe amplitude from -13 to -41 dB with respect to the main one but with the drawback of increasing the width of the main lobe by 30%. This causes the conversion of the sinc function of the PTR into a Gaussian function, shown in Figure 3.1.2-4 and Figure 3.1.2-5, in the frequency domain in FFT p.u. and dB respectively.

However, the output of the receiver is not exactly the one shown in Figure 3.1.2 because we did not consider the Analog-to-Digital Converter. Hence, the output of the receiver block will be the same waveform we concluded above, but sampled. Particularly, in ERS, we have waveforms of 64 samples so a PTR waveform should look like Figure 3.1.3.

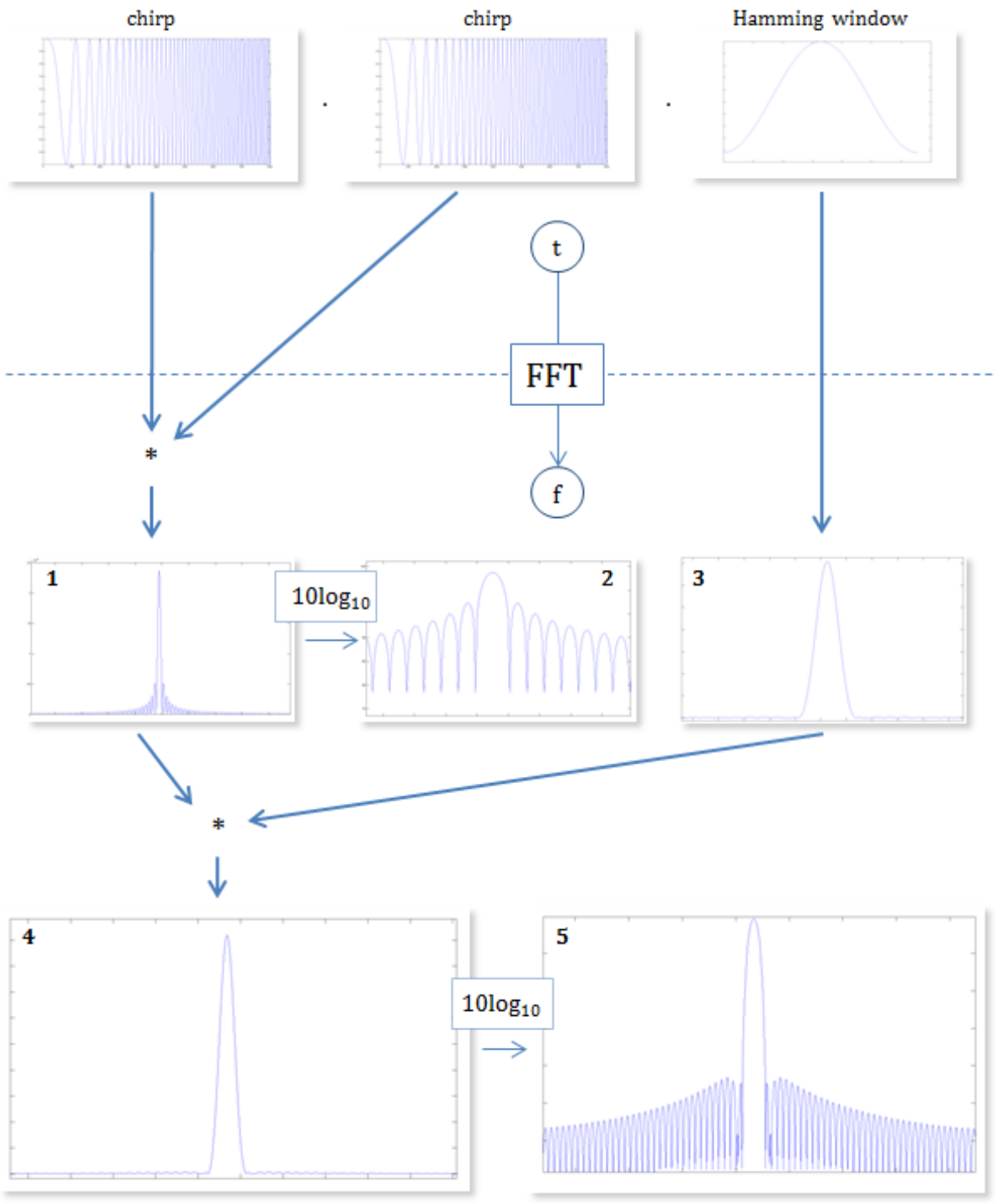


Figure 3.1.2: Steps followed by the original chirp in the PTR chain.

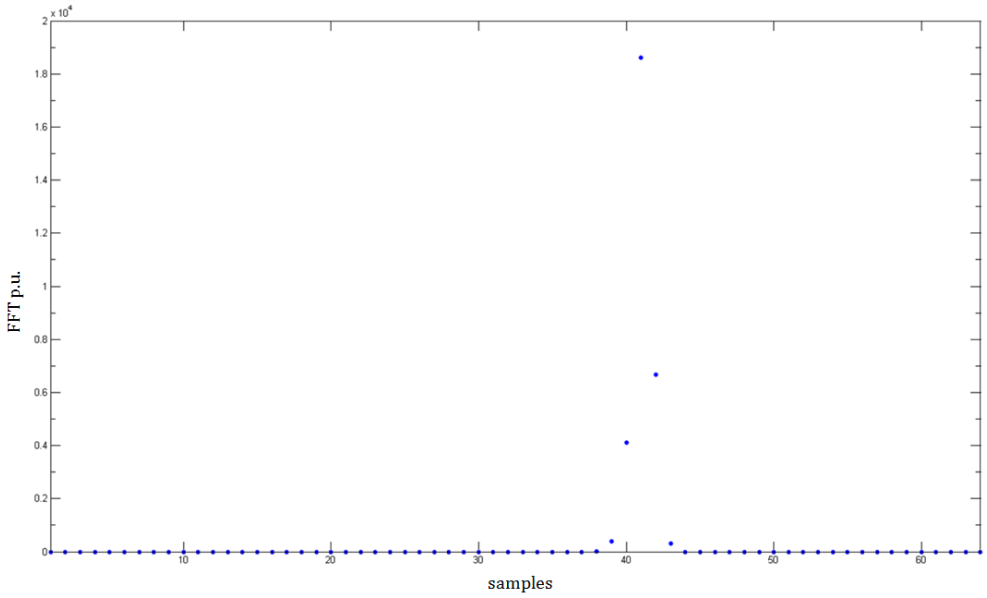


Figure 3.1.3: A PTR ocean-like waveform from ERS-1.

3.2 PTR fitting methods

In order to estimate the PTR, several methods have been proposed, both along the ERS-1 and ERS-2 missions [13]. The fitting methods that we selected for the tests are: the Three-point fitting, the Centre of Gravity, the Simple Gaussian fitting and the Multiple Gaussian fitting.

3.2.1 Three-point fitting

The three-point fitting was ERS method before REAPER. This method is based on the fact that a Gaussian function is the exponential of a parabola. Hence, with a previous logarithmic transformation of the original data, the parameters of the Gaussian function that fits the PTR can be calculated as simple as the calculation of the parameters of a parabola would be[2]. The steps to be done are:

[2]We must note that even the method is called three-point fitting, since the parabola is built with original data points, this method is not a fitting but an interpolation. However, we chose to keep the nomenclature that was used in the ERS mission

1. Find 3 consecutive filter bins with the maximum signal.

2. Let these values be V_1, V_2 and V_3, where V_2 is the maximum and V_1 and V_3 are the adjacent values. f_1, f_2 and f_3 are the index numbers of the filter bins corresponding to V_1, V_2 and V_3 (see Figure 3.2.1).

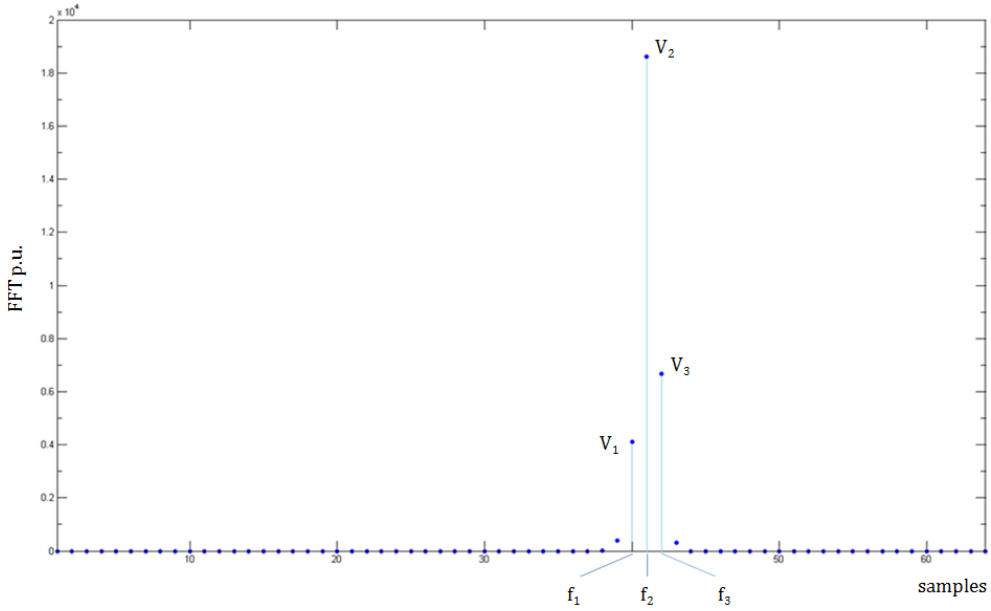


Figure 3.2.1: Selected values from a PTR from ERS-1.

3. Apply the logarithmic transformation:

$$\begin{aligned} z_1 &= lnV_1 \\ z_2 &= lnV_2 \\ z_3 &= lnV_3 \end{aligned} \tag{3.2.1}$$

4. The location of the centre of the pulse $f_{0_{3fit}}$is calculated in terms of filter units:

$$f_{0_{3fit}} = \frac{(z_2 - z_1)(f_1^2 - f_3^2) - (z_3 - z_1)(f_1^2 - f_2^2)}{2\left[(z_3 - z_1)(f_2 - f_1) - (z_2 - z_1)(f_3 - f_1)\right]} \tag{3.2.2}$$

5. The width (W_{3fit}) and the amplitude (A_{3fit}) of the PTR are calculated in

the following way:

$$s = \frac{(z_2 - z_1)}{f_1^2 - f_2^2 + 2\,(f_2 - f_1)\,f_{0_{3fit}}} \tag{3.2.3}$$

$$W_{3fit} = \frac{1}{\sqrt{2s}} \tag{3.2.4}$$

$$A_{3fit} = e^{s(f_1 - f_0)^2 + z_1} \tag{3.2.5}$$

In Figure 3.2.2, the parabola that contains the three logarithmically transformed points z1, z2, z3 is represented together with the position and the amplitude of the maximum of the main lobe of the PTR.

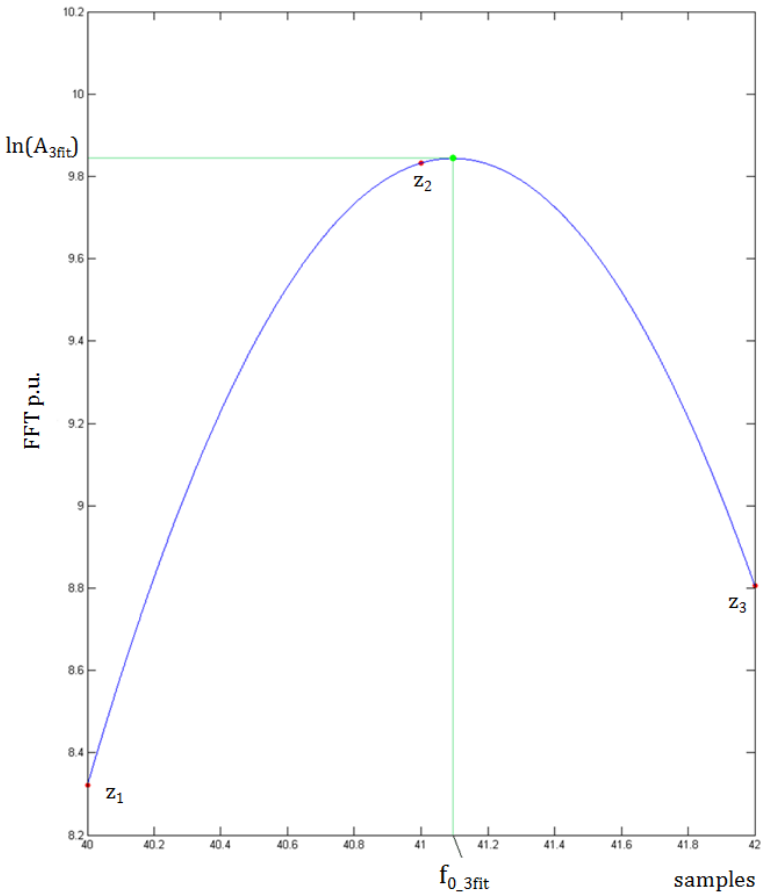


Figure 3.2.2: The points z1, z2 and z3 (in red), the parabola that contains these three points (in blue) and the position and the amplitude of the maximum of the main lobe of the PTR.

3.2.2 Centre of Gravity

This method, as its name states, calculates the centre of gravity of the samples chosen. The position of the maximum of the PTR is then associated to this centre of gravity. This method is simple and only useful for finding the position of the PTR. This means that no width and no amplitude will be calculated. The computation of the centre of gravity will be (see Figure 3.2.3):

1. Dismiss two samples at the beginning and at the end of the waveform to avoid any spurious signals.
2. Set a threshold above the noise floor estimation.
3. Collect the waveform samples above the threshold.
4. Calculate the position of the centre of gravity as stated in Equation 3.2.6:

$$f_{0_{CoG}} = \left(\frac{\sum_{n=2}^{61} V_n \cdot n}{\sum_{n=2}^{61} V_n} \right) \tag{3.2.6}$$

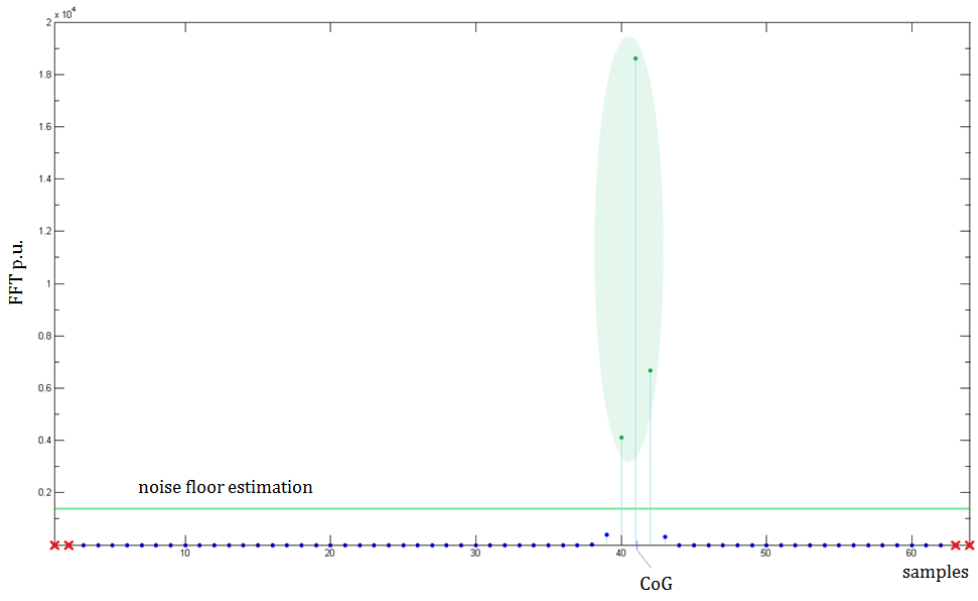


Figure 3.2.3: PTR waveform from ERS-1 together with the samples dismissed with red crosses (step 1), the threshold set (step 2), the samples above this threshold collected in green (step 3) and the resulting position of the PTR (step 4).

This is the simplest way to implement this method. However, other options can be found if, before the computation of the centre of gravity, different interpolations are made. During the study, we have decided to perform several polynomial and cubic splines interpolations. In principle, these interpolations should improve the method, providing more accuracy to the resulting PTR position. This cases will be studied during the tests (see Section 3.3).

3.2.3 Simple Gaussian fitting

This method attempts to reduce the RMS (Root Mean Square) error between a set of N data points $[f_n, V_n]$ with individual standard deviations, σ_n, and a nonlinear function: the Gaussian function defined in Equation 3.2.7.

$$V_n = A_n \cdot e^{-\frac{(f_n - f_0)^2}{2\sigma^2}} \tag{3.2.7}$$

To do so, the *fminsearch* function from Matlab has been used. This function requires an initial guess for the fitted parameters. These are derived from an initial survey of the data set, searching for the maximum value V_{max} and the corresponding position f_{max}. These provide the initial guess for A (amplitude of the PTR) and f (centre of the PTR) respectively. For σ, a constant empirical value shall be used. These are the steps to be followed to implement this method:

1. Select all data pairs satisfying[3]

$$V_n(dB) > V_{max}(dB) - N_{Th} \tag{3.2.8}$$

   where $V_{max}(dB)$ is the maximum amplitude of the PTR in dB and N_{Th} a noise threshold that can be applied to remove possible spurious signals (the default value is 0).

[3]In order to choose the appropriate threshold level, a preliminary study was conducted. Several thresholds from 0 to 40 dB were considered. The final 30 dB threshold was chosen after proving that it was the one that filtered all possible secondary lobes.

2. Fit this reduced data set to a Gaussian curve using the Matlab function *fminsearch* (see Figure 3.2.4).

3. Extract the centre position, $f_{0_{1G}}$, the power, A_{1G} and the standard deviation, σ_{1G} from the Gaussian obtained in the previous step.

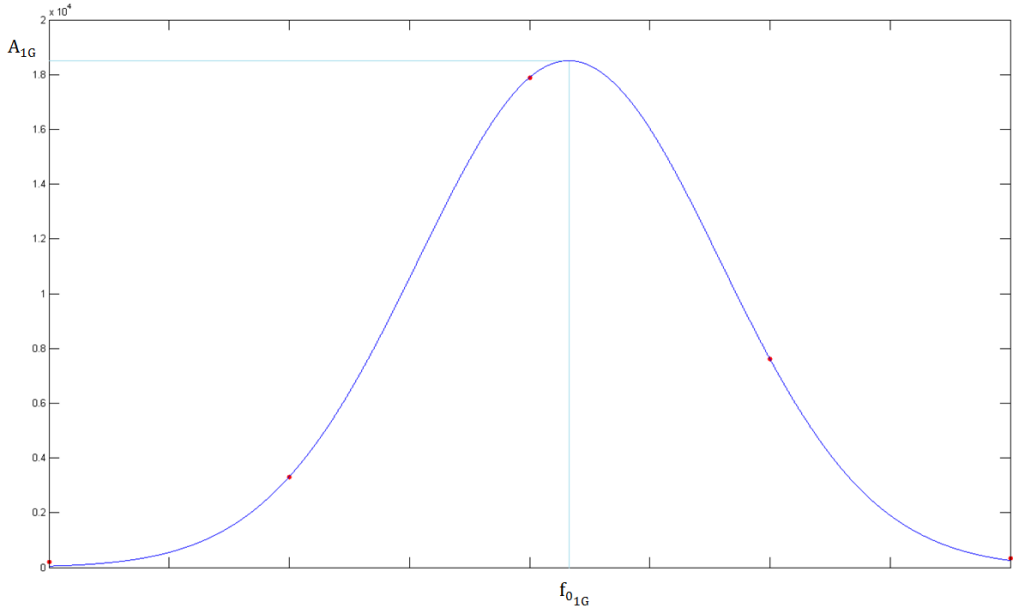


Figure 3.2.4: Fitting of the selected points of a PTR from ERS-1 (red points) with a Gaussian function (blue line).

3.2.4 Multiple Gaussian fitting

The multiple Gaussian fitting is interesting when the PTR shows significant secondary lobes. This is not normally the case when a Hamming weighting is performed on-board, as it occurs in ERS and EnviSat. However, in old studies was demonstrated that, in the case of ERS-1, a fitting with three Gaussians provided better results. This was because of spurious signals that were shown as secondary lobes. Therefore, the multiple Gaussian fitting will be performed by applying a fitting with three Gaussians to the PTR waveform. The steps to be done are the following:

1. Perform the same steps as in Section 3.2.3 (Simple Gaussian fitting), this time with a three-Gaussian function, as shown in the equation below:

$$V_n = A_{n_{left}} \cdot e^{-\frac{\left(f_n - f_{0_{left}}\right)^2}{2\sigma^2_{left}}} + A_{n_{centre}} \cdot e^{-\frac{\left(f_n - f_{0_{centre}}\right)^2}{2\sigma^2_{centre}}} + A_{n_{right}} \cdot e^{-\frac{\left(f_n - f_{0_{right}}\right)^2}{2\sigma^2_{right}}} \tag{3.2.9}$$

where:

- V_n is the amplitude of the FFT sample in filter n,
- f_n is the FFT filter number of PTR point n,
- f_0 is the centre of the PTR and
- σ is the standard deviation.

The result of fitting this three-Gaussian function to a hypothetic PTR with significant secondary lobes at both sides of the main lobe is shown in Figure 3.2.5.

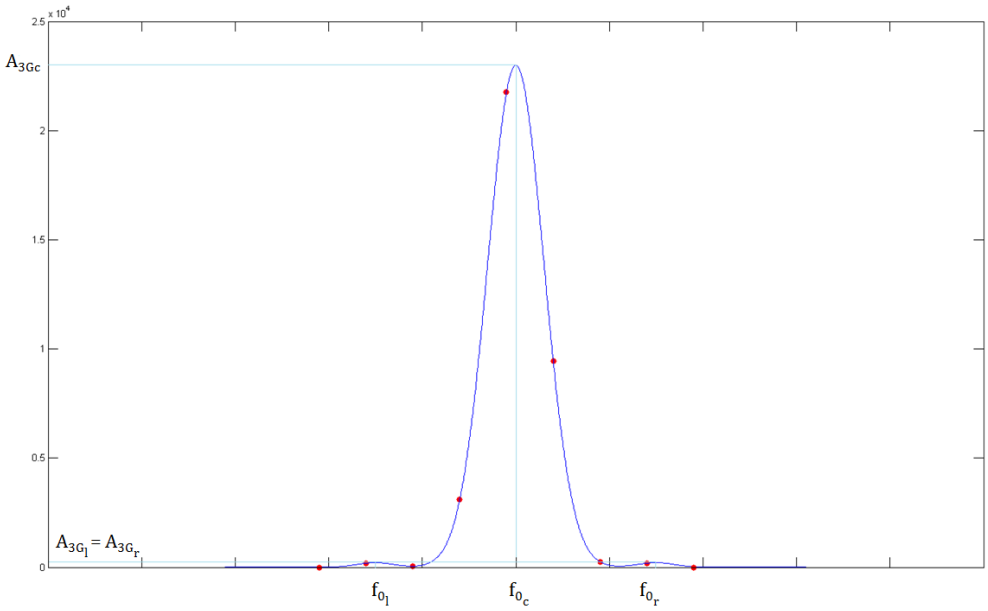


Figure 3.2.5: Fitting of a PTR-like waveform with secondary lobes (red points) with three Gaussians (blue line).

3.3 Testing

The main goal of this section is to conclude which is the best fitting method for the PTR and thus which method will be implemented to reprocess the ERS data. To do so, we will carry out three stages of tests.

The first one, the short-term tests stage, includes both specific key tests based on theory and individual, real PTR tests. This means that, in the first case, the input data to fit will be specifically prepared data instead of real PTR waveforms. This will let us know which method would work better in an ideal case. In the second case, however, some tests will be done with real PTR waveforms. Even these two cases use different input data, they have one common purpose and that is to check how powerful the methods are when fitting different types of data, how capable they are to adjust to a given number of points.

On the other hand, the second stage of tests is based on mid-term input data. That is, all the PTR waveforms contained in one orbit. This stage aims to observe which method obtains better results when calculating the trend of the different parameters of the PTR during a longer period of time. Afterwards, according to the results, we will decide whether the precision of parameters estimation is preferred rather than a more continuous trend of them, or a combination of both options. Before performing the long-term tests, some methods will be discarded.

The last part of this section is the long-term tests stage. These tests will use a whole cycle of data to evaluate the methods that best performed in the mid-term tests.

Before carrying out all the tests (except for the ideal case of the short-term tests), we had to build a module that read and extracted the information from the input files. The structure of these input files and the way to extract them was defined in [14]. However, there were several confusing statements as well as significant errors and our job was also to issue new, updated documentation [15, 16], as an outcome of the REAPER project. Some of this work can be found in Chapter 5.

3.3.1 Short-term tests

As we already mentioned above, this stage of tests is divided in two parts: the ideal and the real case.

3.3.1.1 Ideal case

This first part of the short-term tests is based on the theoretical study we already performed. That means the input PTR data will be a sampled Gaussian function, which will vary depending on the mode we are working in order to have a more realistic scenario[4]. The procedure of the following tests is to generate a Gaussian function with specific parameters, sample it and run the different methods with this input data. Then we will see how the parameters of the resulting fitted function differ from the original ones and hence conclude which method works best.

The chosen Gaussian functions are plotted in Figure 3.3.1.

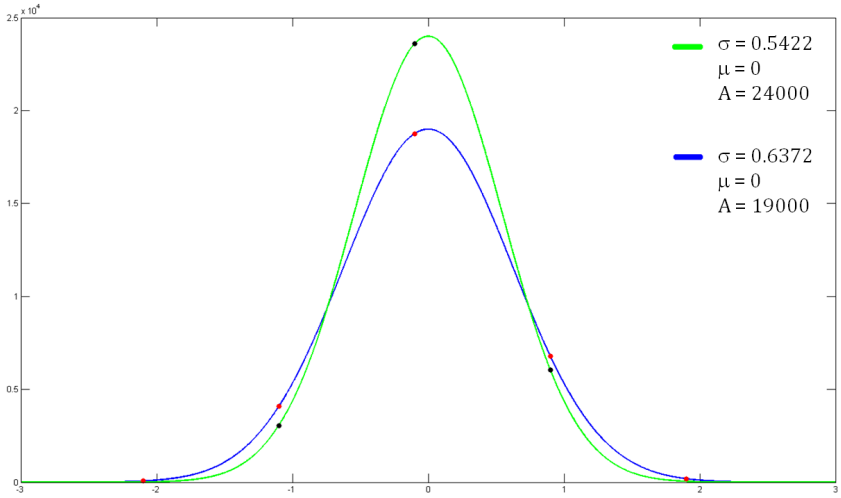


Figure 3.3.1: The generated ice (green) and ocean (blue) Gaussian functions, with their corresponding parameters (in FFT power units), together with the samples taken from them (points in black and red, respectively).

[4]With the parameters chosen, the Gaussian functions generated are very similar to ice and ocean-like PTR waveforms.

As we can see in Figure 3.3.1, the generated Gaussian functions have been sampled. With this data, the results obtained with the different methods are shown in Tables 3.3.1, 3.3.2, 3.3.3 and 3.3.4.

Note: f_0 and σ are expressed in samples. A is expressed in FFT power units.

Three-point fitting

Parameter	Ice	Ocean
f_0	$-5.7732 \cdot 10^{-15}$	$1.7764 \cdot 10^{-15}$
σ	$0.5422 - 2.5535 \cdot 10^{-15}$	$0.6273 + 7.7716 \cdot 10^{-16}$
A	$24000 - 1.4552 \cdot 10^{-11}$	$19000 - 7.2760 \cdot 10^{-12}$

Table 3.3.1: Parameters obtained with the Three-point fitting method, for both ice and ocean modes.

Centre of Gravity

Interpolation	Parameter	Ice	Ocean
No interpolation	f_0	-0.0087	-0.0013
Polynomial (grade 4)	f_0	-0.0530	-0.0374
Cubic splines	f_0	-0.0608	-0.0304

Table 3.3.2: Results obtained with the most signficant choices for the CoG, for both ice and ocean modes.

Simple Gaussian fitting

Parameter	Ice	Ocean
f_0	$1.3323 \cdot 10^{-15}$	$1.7764 \cdot 10^{-15}$
σ	0.5422	$0.6273 + 1.1102 \cdot 10^{-16}$
A	24000	$19000 - 3.6380 \cdot 10^{-12}$

Table 3.3.3: Parameters of the Gaussian function obtained with the Simple Gaussian fitting method, for both ice and ocean modes.

Multiple Gaussian fitting

Parameter	Ice	Ocean
f_0	$-2.3793 \cdot 10^{-6}$	$-3.9796 \cdot 10^{-11}$
σ	$0.5422 - 3.4534 \cdot 10^{-6}$	$0.6273 - 4.088 \cdot 10^{-11}$
A	23999.9792	$19000 - 8.0909 \cdot 10^{-8}$

Table 3.3.4: Parameters of the main Gaussian function obtained with the Multiple Gaussian fitting method, for both ice and ocean modes.

So, for now, we can appreciate that the Three-point, the Simple Gaussian and the Multiple Gaussian fittings work great and practically obtain the exact value of the parameters of the original Gaussian function. However, in the ice case, the last ones lowers a bit its accuracy, but it still obtains remarkable results considering the method tries to fit three Gaussian functions to only one.

On the other hand, the results obtained with the Centre of Gravity are not so close to the original ones. Moreover, the interpolations performed do not improve the method capabilities as their results differ more from the original parameters than the ones obtained without any interpolation.

As far as the PTR position estimation is concerned, it is a good idea to translate the errors given above into height in order to better understand them. The methods that work better (that is, the Three-point, the Simple Gaussian and the Multiple Gaussian fittings) have errors of the order of 0.02 nm in the worst case and of $8 \cdot 10^{-7}$ nm in the best case. These are great results because they perfectly meet the precision needed (only measures allowed are those below 1 mm error). On the other hand, even in the best case, the CoG with no interpolation achieves the required accuracy (0.6 mm approximately in the ocean case), the other cases have errors of more than 10 mm. Thus, even the CoG methods will be tested in the real case of this section, they will be definitely discarded for the mid and long-term tests.

3.3.1.2 Real case

The objective of this second part of the short-term tests is the same as in the ideal case: see how capable is each method to fit a single PTR waveform. Nevertheless, since we don't have the original parameters of the theoretical Gaussian function as a reference, the conclusion will not be so objective.

The execution of the tests, in both modes, returns the following results.

Ice mode In ice mode, the PTR waveforms usually only have 3 samples (see Figure 3.3.2).

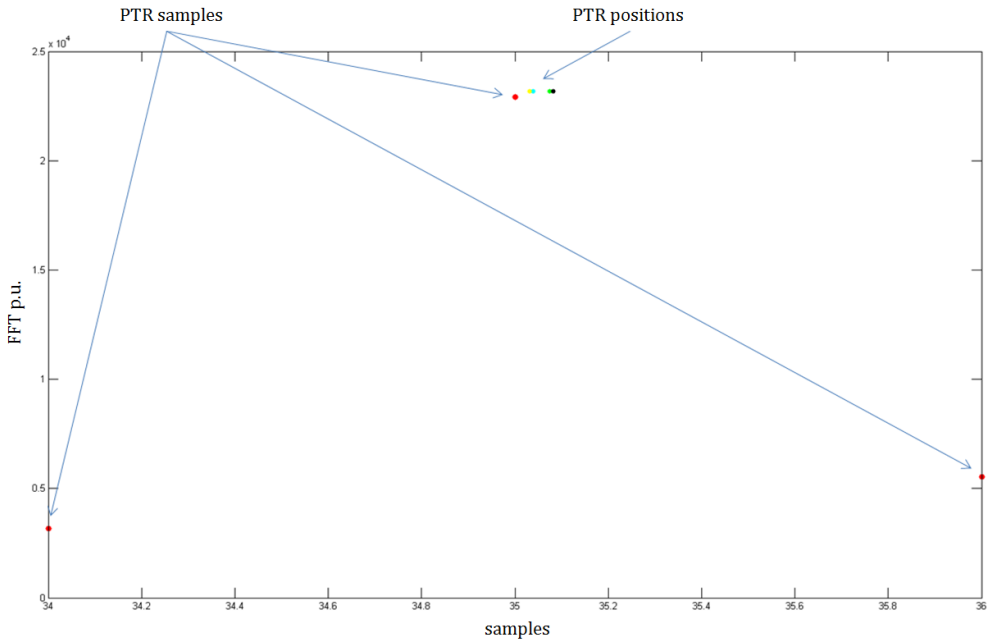


Figure 3.3.2: An ice PTR waveform from ERS-1 (in red) with the estimated PTR positions.

If we zoom in Figure 3.3.2 (see Figure 3.3.3), we observe the results can be separated in two different groups. The first one is formed by the positions estimated by the three Centre of Gravity methods and the second one by the positions of the rest of the methods.

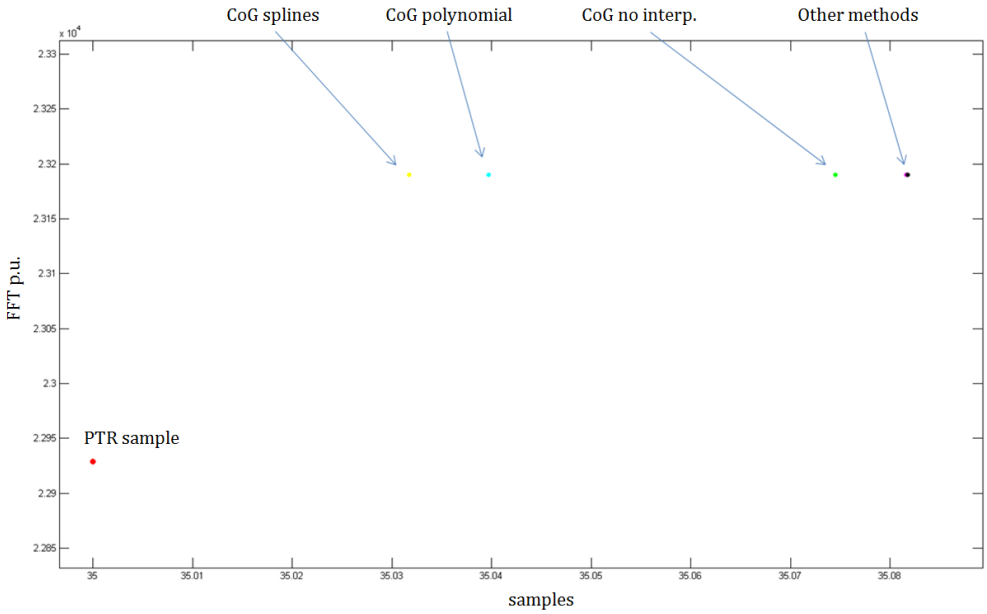


Figure 3.3.3: Zoom of Figure 3.3.2

Moreover, if we zoom again in oder to see the other estimations clearer (Figure 3.3.4), we can observe that the positions obtained with the Three-point fitting and the Simple Gaussian fitting coincide.

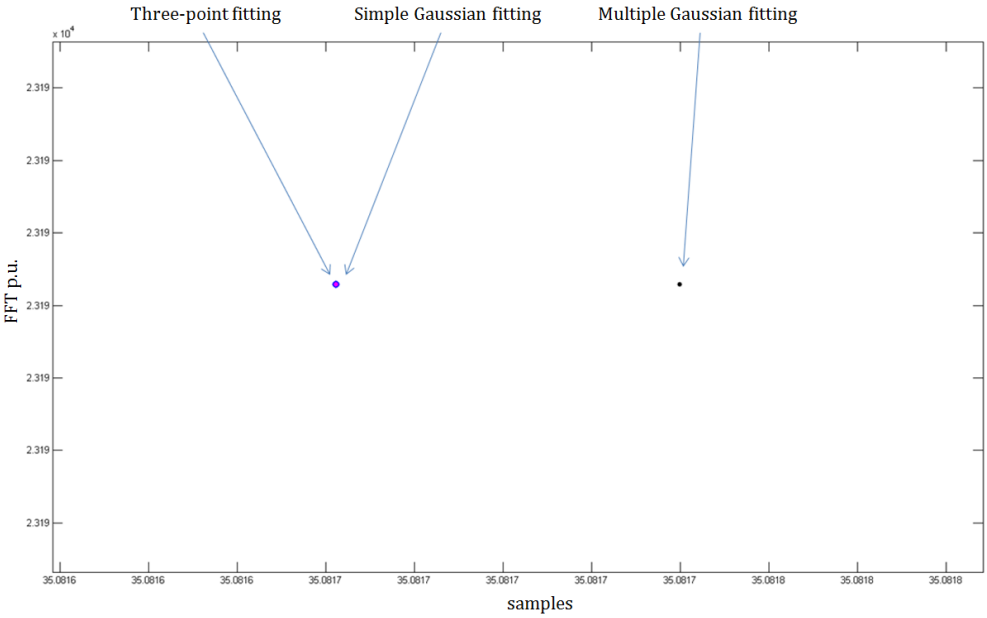


Figure 3.3.4: Zoom of Figure 3.3.3

The values obtained with the different methods are shown in Tables 3.3.5 and 3.3.6.

Method	**CoG No Int.**	**CoG Poly**	**CoG Splines**
$\mathbf{f_0}$	35.074515	35.039691	35.031716

Table 3.3.5: PTR positions of the three Centre of Gravity methods.

Method	**Three-point**	**Simple Gauss**	**Multi Gauss**
$\mathbf{f_0}$	35.081662	35.081662	35.081739

Table 3.3.6: PTR positions of the other three methods.

Ocean mode In the ocean mode, ideally, the PTR waveforms have 5 samples different from zero. However, in ERS-1, as [**13**] states, the PTR waveforms may have spurious signals that can be shown as secondary lobes. We have also seen this behaviour through the mission.

Then, since the secondary lobes are only present in one side of the main lobe, these PTR waveforms have 6 samples different from zero. Because of this, we have considered a modification of the Multiple Gaussian fitting by reducing its number of Gaussians to two.

In order to decide if the secondary lobes affect the performance of the fittings, we will carry out two differents tests: the first one without secondary lobes and the second one with secondary lobes.

Ocean mode without secondary lobes This first case represents a nominal example, with no clear secondary lobes and where no deletion of PTR samples is performed.

As we can see in Figure 3.3.5, all the fitting methods work fine[5]. We can also observe that, with the correct seed, the Multiple Gaussian fittings do not include secondary lobes when they are not present in the original data.

[5]Note that the CoG methods are not displayed since they are not fitting functions.

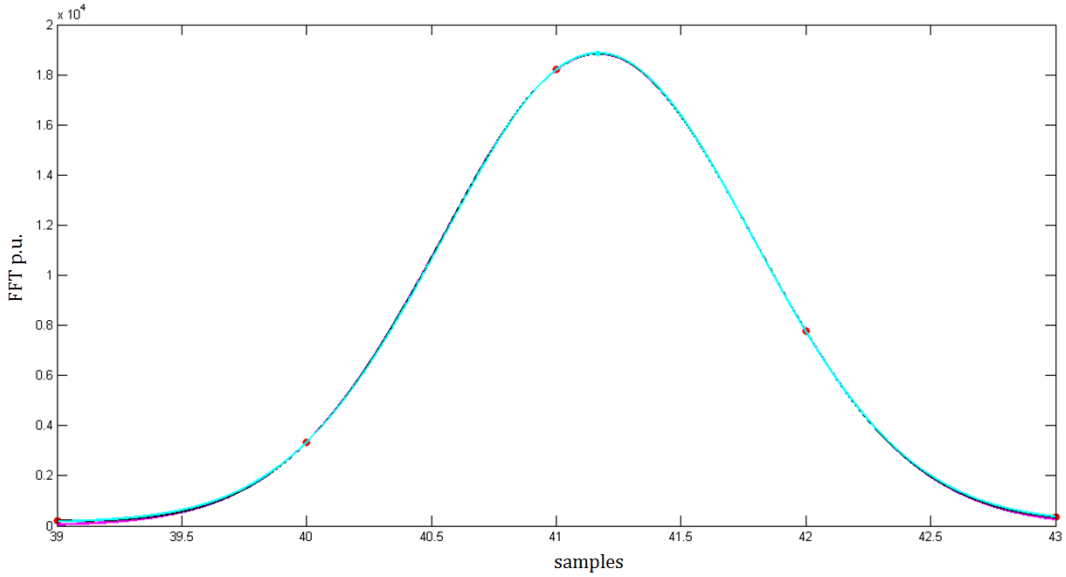


Figure 3.3.5: The output functions of the methods (the Three-point fitting in blue, the Simple Gaussian in pink, the 3-Gaussian in black and the 2-Gaussian in cyan).

Moreover, we see that both the Three-point (actually an interpolation) and the Simple Gaussian fittings do not pass through the original data point (Figure 3.3.6). This might happen because of some interference that distorts the shape of the PTR even it is not shown as a clear secondary lobe.

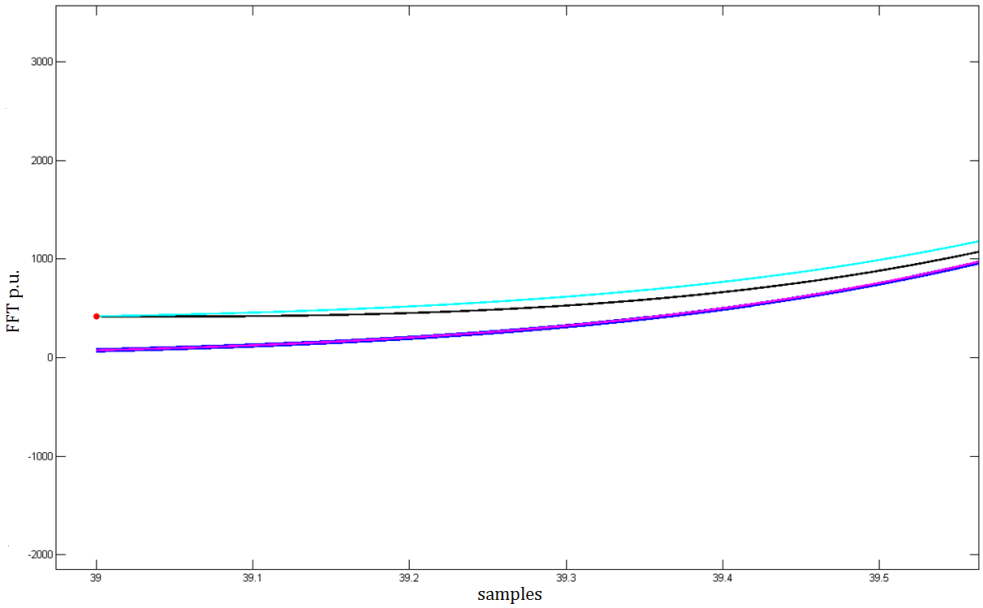


Figure 3.3.6: Zoom of Figure 3.3.5 around sample 39.

Regarding the PTR position estimations, Figure 3.3.7 shows the obtained results:

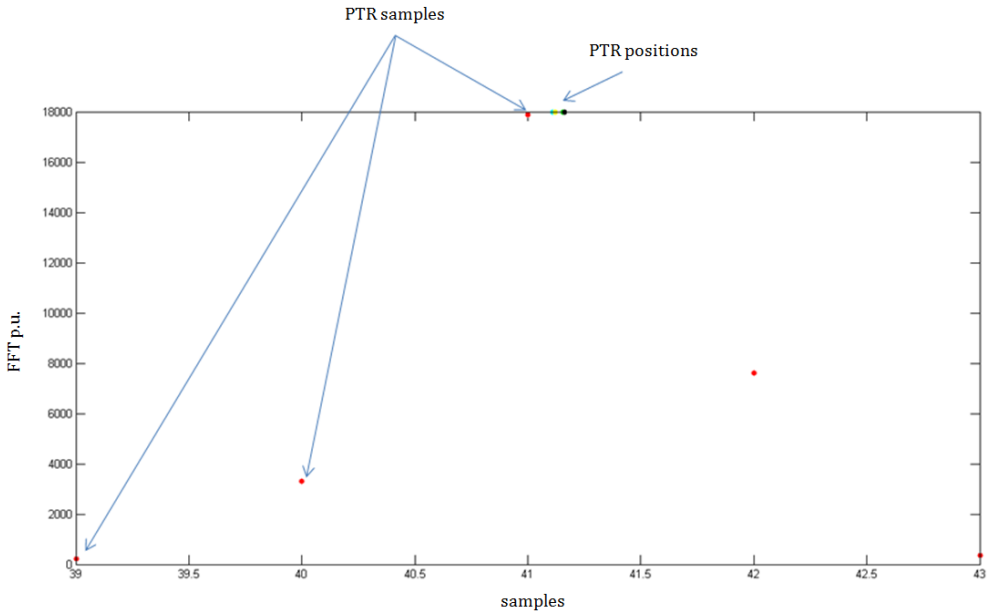


Figure 3.3.7: An ocean PTR waveform from ERS-1 (in red) with the estimated PTR positions.

If we zoom in Figure 3.3.7, we can see the estimated PTR positions clearer (Figure 3.3.8). Again, we can clearly distinguish two different groups.

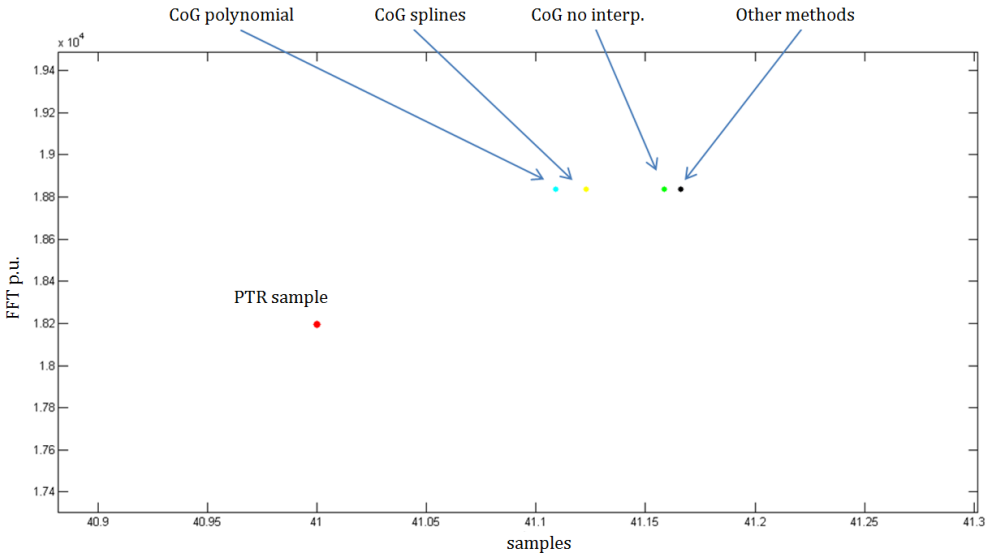


Figure 3.3.8: Zoom of Figure 3.3.7.

In this second scenario, the two groups observed in the ice mode can also be distinguished. Figure 3.3.9 shows the PTR position estimated by the other three methods.

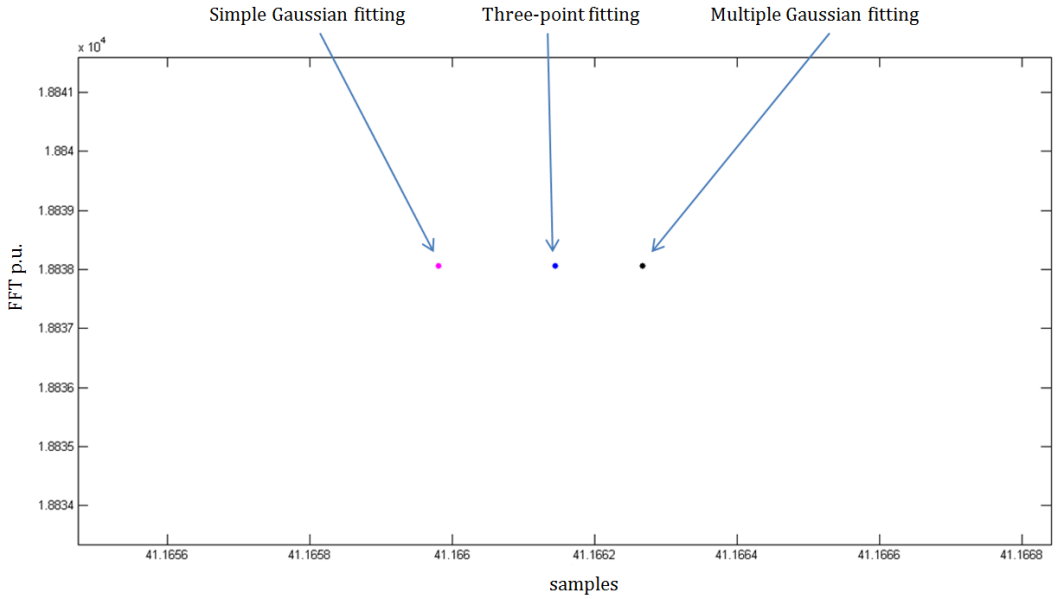


Figure 3.3.9: Zoom of Figure 3.3.8.

Tables 3.3.7 and 3.3.8 show the exact values represented in the figures above:

Method	**CoG No Int.**	**CoG Poly**	**CoG Splines**
$\mathbf{f_0}$	41.158431	41.109157	41.122979

Table 3.3.7: PTR positions of the three Centre of Gravity methods.

Method	**Three-point**	**Simple Gauss**	**Multi Gauss**
$\mathbf{f_0}$	41.166144	41.165979	41.166267

Table 3.3.8: PTR positions of the other three methods.

Even the methods output different results for the PTR position:

- the difference between the Simple Gaussian and the Multiple Gaussian fittings is about 0.13 mm;
- the difference between the Simple Gaussian and the Three-point fittings is around 0.075 mm.

Ocean mode with secondary lobes This second case, which is only present in ERS-1, will show us how the different fitting methods behave in the presence of secondary lobes as well as when we delete these secondary lobes from the original data.

In this scenario, we will not run the CoG methods because, as seen in all previous tests, they do not provide good results. On the other hand, the Three-point fitting will not be evaluated -although present in the plots- because it does not account for the presence of secondary lobes and thus it would provide the same results in both examples.

The results of the fittings are shown in Figure 3.3.10.

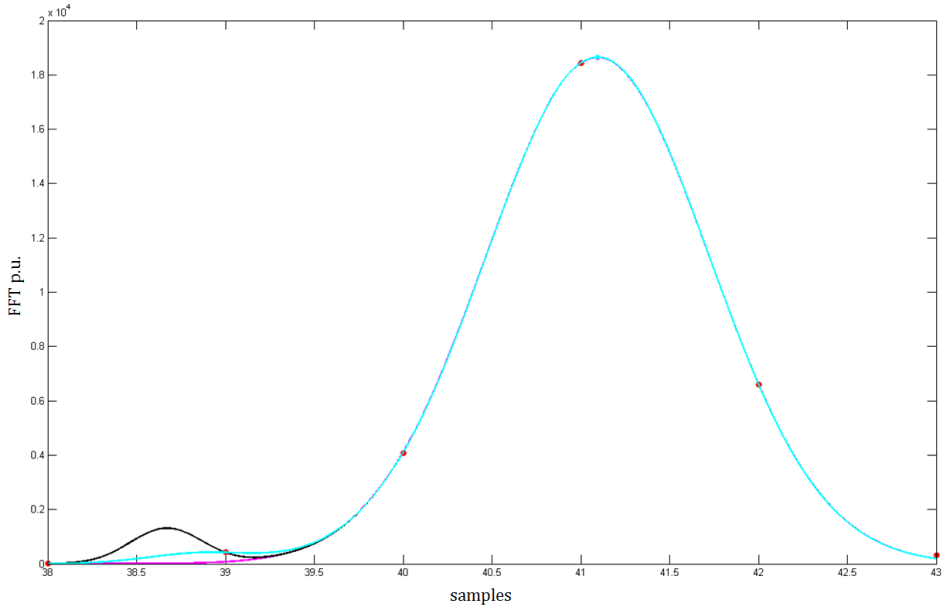


Figure 3.3.10: A sampled PTR waveform with secondary lobes (in red) together with the results of the fitting methods. As we can observe, the two Multiple Gaussian fittings place the secondary lobe in different places.

In figures 3.3.11 and 3.3.12 we can see how close the fitted functions to the PTR samples are. As we zoomed in two of the three samples of higher amplitude, we observe that the Three-point fitting exactly passes through the points. As far as the Gaussian fittings are concerned, the Multiple Gaussian fittings are closer to the PTR sample than the Simple Gaussian fitting is. This confirms what we had already seen in the previous case: the Simple Gaussian fitting is the method that

least fits the original PTR samples. However, the Multiple Gaussian fittings are very seed dependent.

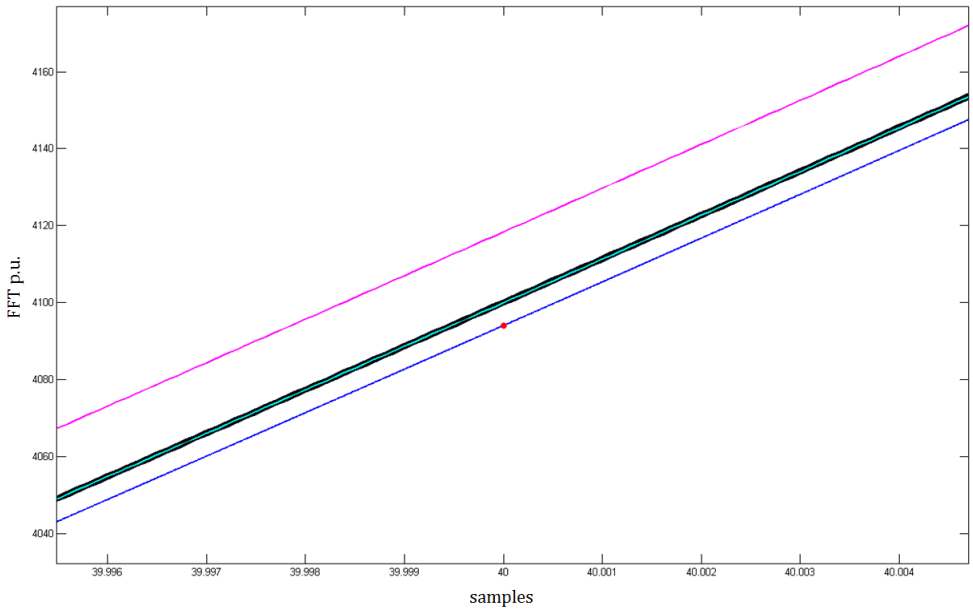


Figure 3.3.11: Zoom of Figure 3.3.10 at the sample in x = 40.

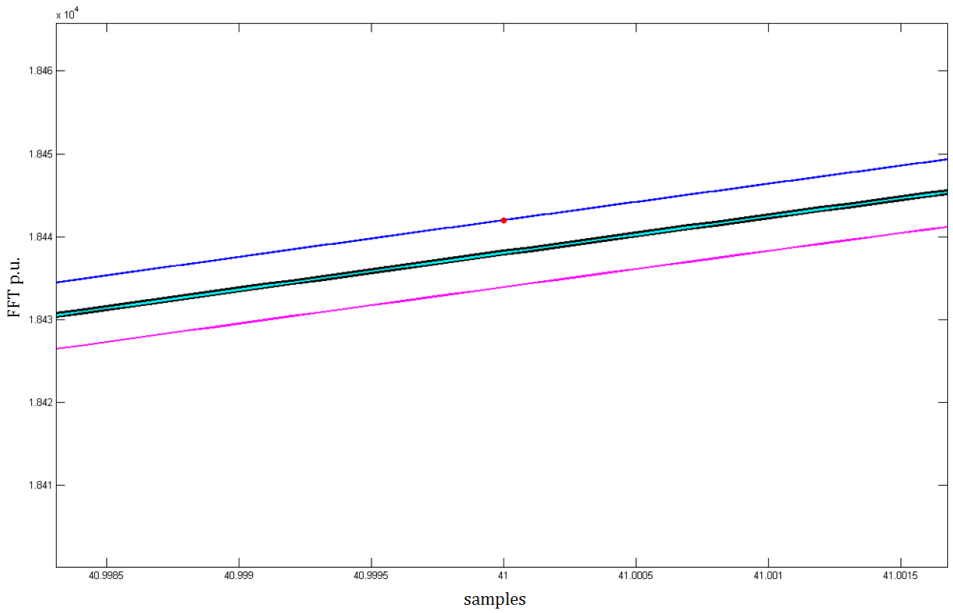


Figure 3.3.12: Zoom of Figure 3.3.10 at the sample in x = 41.

On top of that, the purpose of this test is to observe how the methods estimation change with or without the presence of a secondary lobe. To do this, we will choose

a PTR waveform with a secondary lobe, run the test, remove the secondary lobe and then run the test again.

The results of the fittings are the following:

Method	Simple Gauss	3 Gauss	2 Gauss
f_0 (no secondary lobe)	41.094205	41.096211	41.101043
f_0(with secondary lobe)	41.094105	41.094773	41.094783

Table 3.3.9: PTR positions (in samples) when a secondary lobe is or is not present.

Method	Simple Gauss	3 Gauss	2 Gauss
σ (no secondary lobe)	0.62966	0.62709	0.61821
σ (with secondary lobe)	0.62966	0.62897	0.62897

Table 3.3.10: PTR width (in samples) when a secondary lobe is or is not present.

As we can see, the Simple Gaussian fitting is the one that experiences fewer changes with the presence (or absence) of a secondary lobe: the width remains unchanged and the PTR position only varies 0.0001 samples (0.044 mm). On the other hand, the Multiple Gaussian fittings seem to be more affected and have bigger changes in both parameters.

Even the methods output different results for the PTR position, the difference between the Simple Gaussian and the Multiple Gaussian fittings is about 0.3 mm.

3.3.2 Mid-term tests

These tests will manage data from a whole orbit. This means that we will extract and process all the PTR data stored in an ERAC[6] file. From the results that we will obtain, as the computational time required afterwards will raise significantly, our intention is to choose the best two fitting methods before we perform the long-term tests. The two parameters that will be studied will be the PTR time delay, for having a direct impact on the range estimation, and the PTR width, for being highly involved in the SWH[7] calculation.

However, in order to make sure our selection is good, instead of only processing one ERAC, we will process four ERACs: two of ERS-1 and two of ERS-2. Table 3.3.11 shows the start and end dates of the ERACs that we selected.

#ERAC	Satellite	Start date	End date
1	ERS-1	25.03.1996 05:33.08	25.03.1996 07:28.54
2	ERS-1	01.04.1996 17:04.15	01.04.1996 18:43.50
3	ERS-2	06.04.1996 16:39.09	06.04.1996 18:18.17
4	ERS-2	24.04.1996 23:40.59	25.04.1996 01:39.02

Table 3.3.11: Start and end dates of the input files.

Apart from running two different files per satellite, we will study both the evolution of the PTR position and the PTR width, in both ocean and ice modes. Furthermore, as the input data may contain spurious signals, we also considered including an extra Simple Gaussian fitting so then we are able to test the algorithm when it is taking into account possible secondary lobes and when it is deleting them.

Note: All the following plots have the same legend.

Three-point fitting | Simple Gaussian fitting with Th | Multiple Gaussian fitting (2)
Simple Gaussian fitting | Multiple Gaussian fitting (3)

[6]ERAC is the name that ERS L0 products receive.
[7]Significant Wave Height.

3.3.2.1 ERS-1 ice mode

- ERAC #1

In this first example, both PTR position (Figure 3.3.13) and width (Figure 3.3.14) estimations from the Three-point and the two Simple Gaussian fittings coincide. On the other hand, the estimations from the two Multiple Gaussian fittings, even though they are close to the others, they present some outliers. The observed undulations are due to some orbital factors, such as temperature.

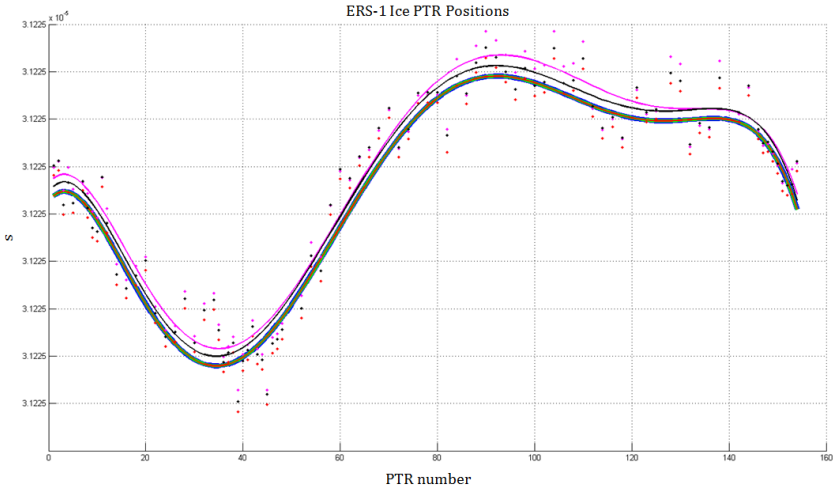


Figure 3.3.13: ERS-1 PTR positions in ice mode together with a polynomial fitting of their trend.

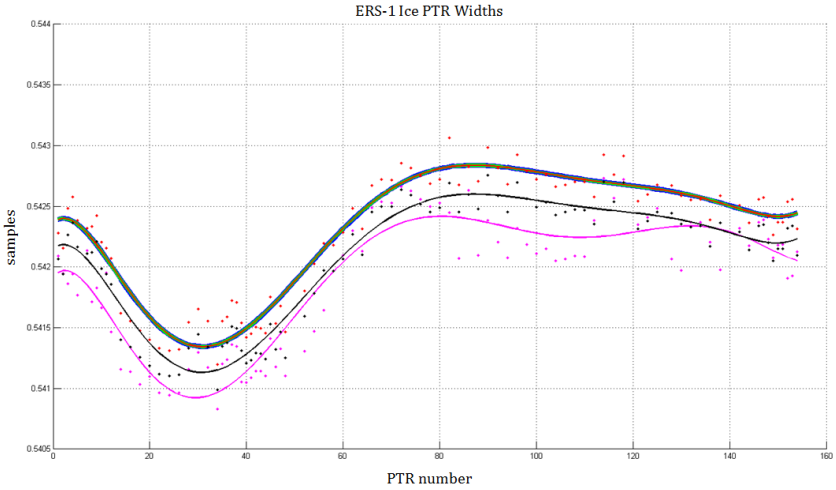


Figure 3.3.14: ERS-1 PTR widths in ice mode together with a polynomial fitting of their trend.

- ERAC #2

In this second example (Figure 3.3.15) we see the same behaviour as in Figure 3.3.13. On the other hand, though, the PTR widths (Figure 3.3.16) from the two Multiple Gaussian fittings differ a lot from the other methods.

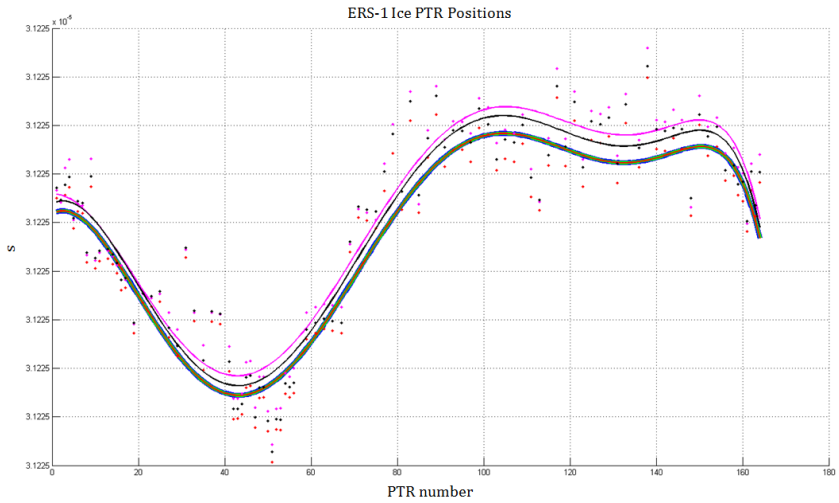


Figure 3.3.15: ERS-1 PTR positions in ice mode together with the polynomial fitting of their trend.

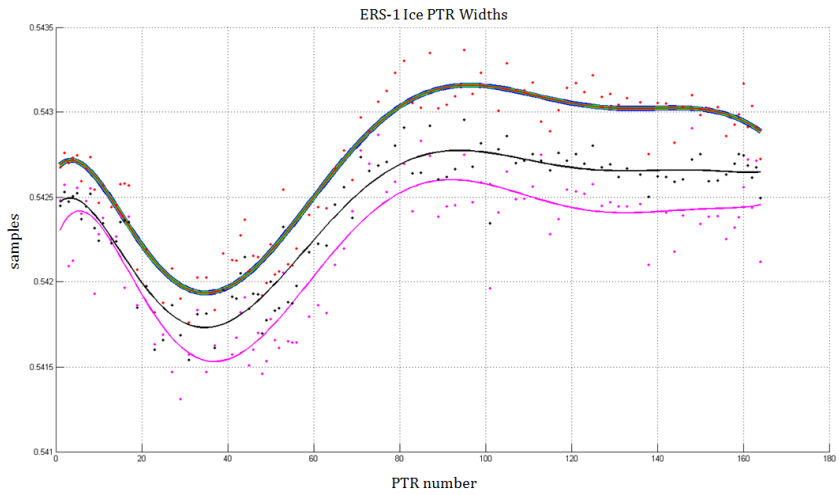


Figure 3.3.16: ERS-1 PTR widths in ice mode together with the polynomial fitting of their trend.

3.3.2.2 ERS-1 ocean mode

- ERAC #1

In ocean mode, we obtained huge outliers from the two Multiple Gaussian methods. Even we removed the farthest ones, we still see that the polynomial fittings of their trends cannot follow any logical pattern (Figure 3.3.17 and Figure 3.3.19). Then, if we remove these erratic results, we observe that the Three-point and the two Simple Gaussian fittings do not overlap as in the ice mode.

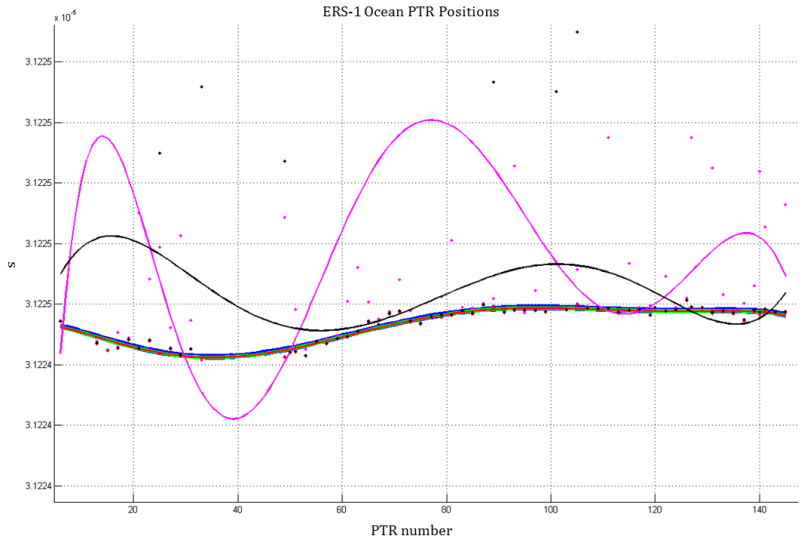


Figure 3.3.17: Zoom of the original figure with ERS-1 PTR positions for ocean mode.

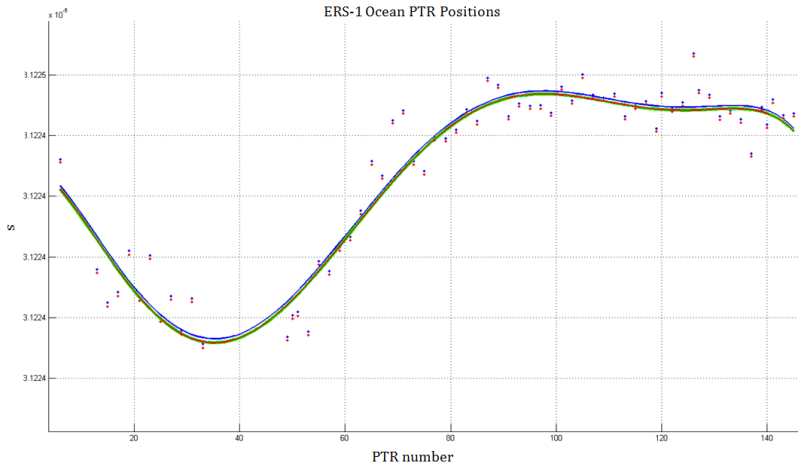


Figure 3.3.18: A zoom of Figure 3.3.17. The two Simple Gaussian fittings have what seems a constant offset with respect to the Three-point fitting.

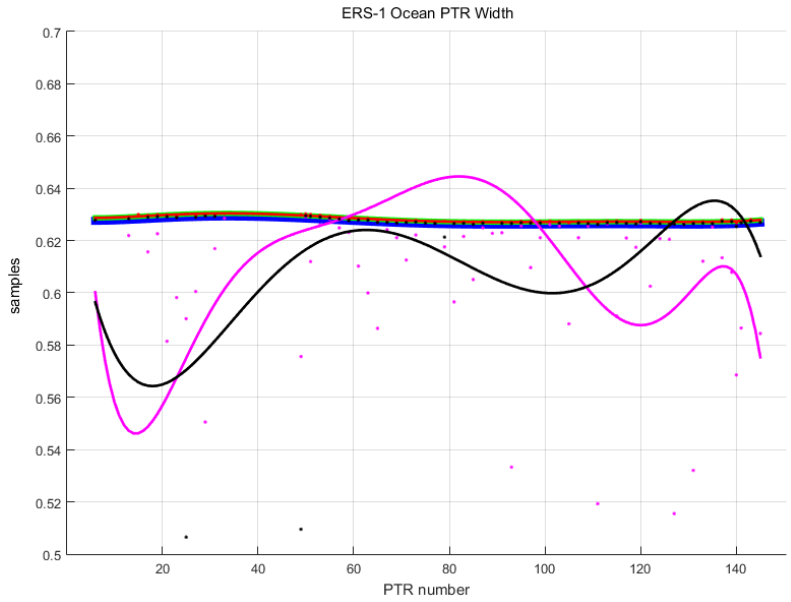


Figure 3.3.19: Zoom of the original figure. A closer view of the outliers.

In Figure 3.3.20, we can observe that in the width estimations there is also an offset between the Simple Gaussian fittings and the Three-point fitting, which is equivalent to roughly 0.5 mm.

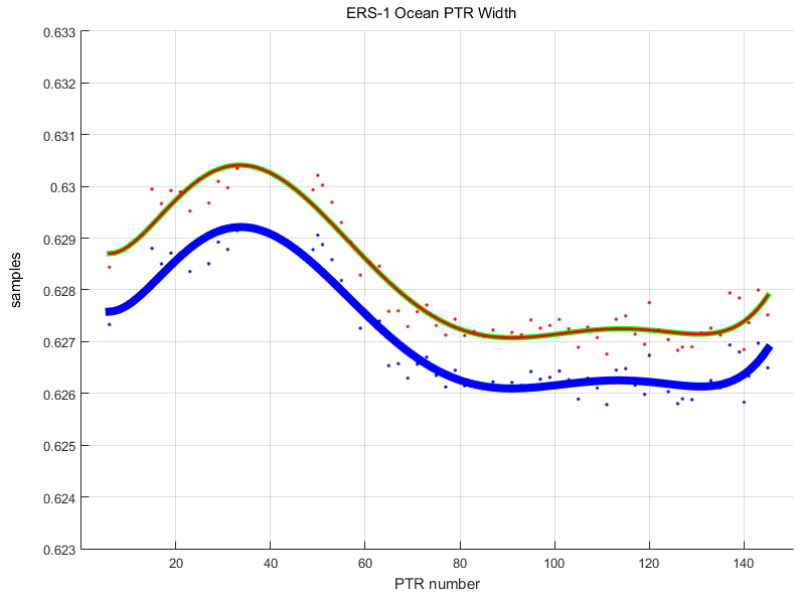


Figure 3.3.20: Zoom of Figure 3.3.19. As we did before, we removed the results of the Multiple Gaussian fittings. Here the bias is observed clearer.

- ERAC #2

In this second example, we even have more outliers than in the first case. This can be seen both in Figure 3.3.21 and Figure 3.3.23.

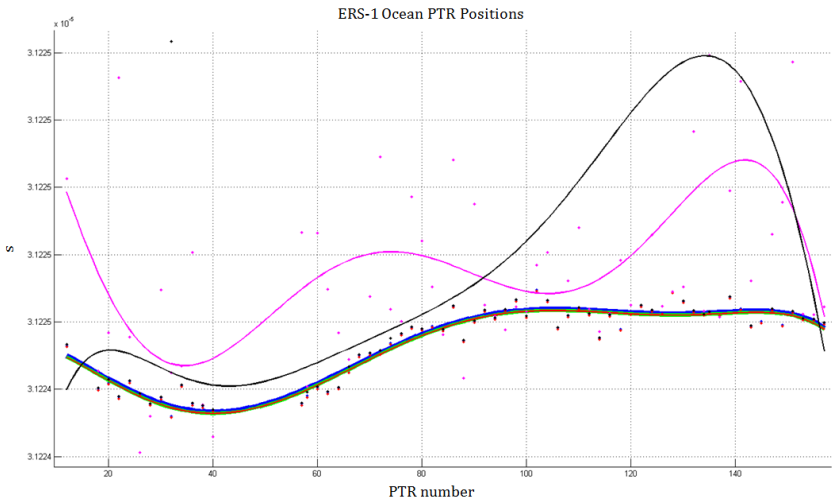


Figure 3.3.21: Zoom of the original figure of PTR positions for ocean mode.

Again, when we remove the results of he Multiple Gaussian fittings, we observe that there is an offset between the values of the two Simple Gaussian and the Three-point fittings, which is equivalent to slightly less than 0.1 mm for the PTR positions (see Figure 3.3.22).

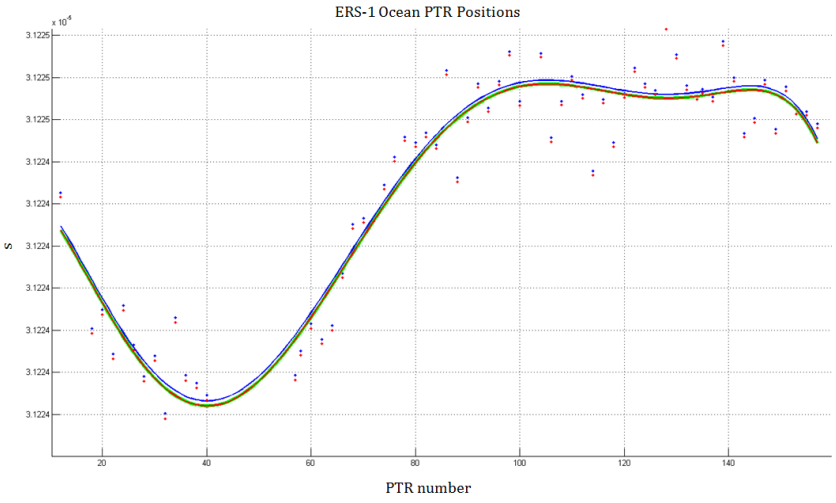


Figure 3.3.22: Zoom of Figure 3.3.21. ERS-1 PTR positions from the Three-point and the two Simple Gaussian fittings in ocean mode.

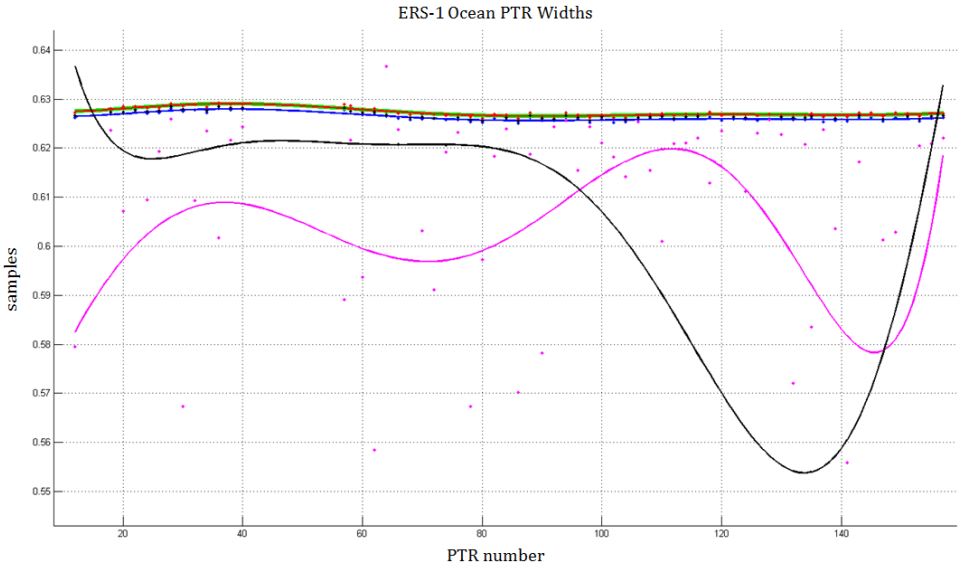


Figure 3.3.23: Zoom of the original figure of the PTR widths.

In the case of the PTR width estimations, the offset between the Three-point fitting and the two Simple Gaussian fittings is roughly equivalent to 0.5 mm.

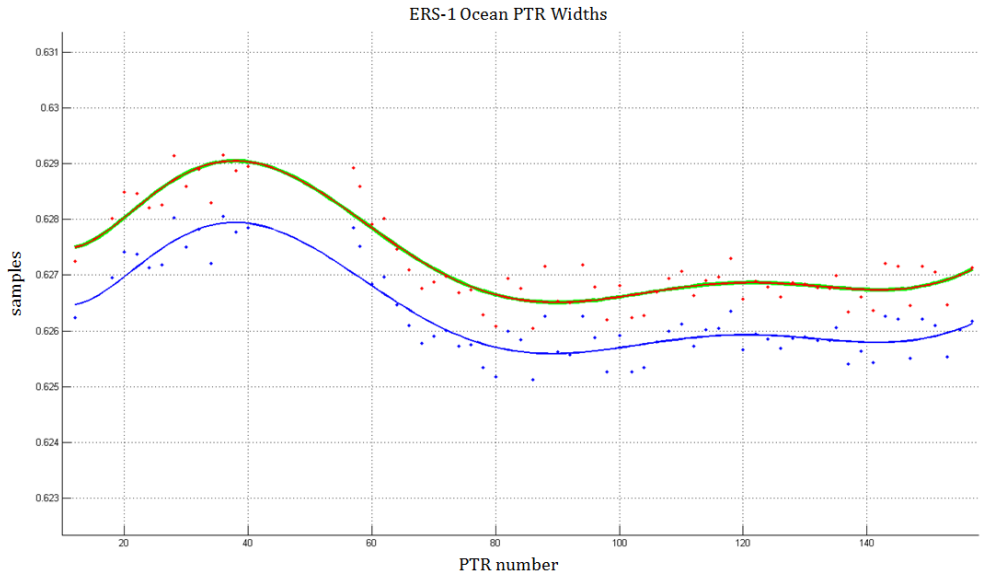


Figure 3.3.24: Zoom of Figure 3.3.23.

3.3.2.3 ERS-2 ice mode

- ERAC #1

In the first example of ERS-2 ice mode, we observe the same behaviour as in ERS-1 ice mode: the Three-point and the Simple Gaussian fittings coincide, both for the PTR position (Figure 3.3.25) and width estimations (Figure 3.3.26). On the other hand, the 3-Gaussian fitting seems to have a bias with respect to them, while the 2-Gaussian fitting now follows a totally different trend due to too many outliers.

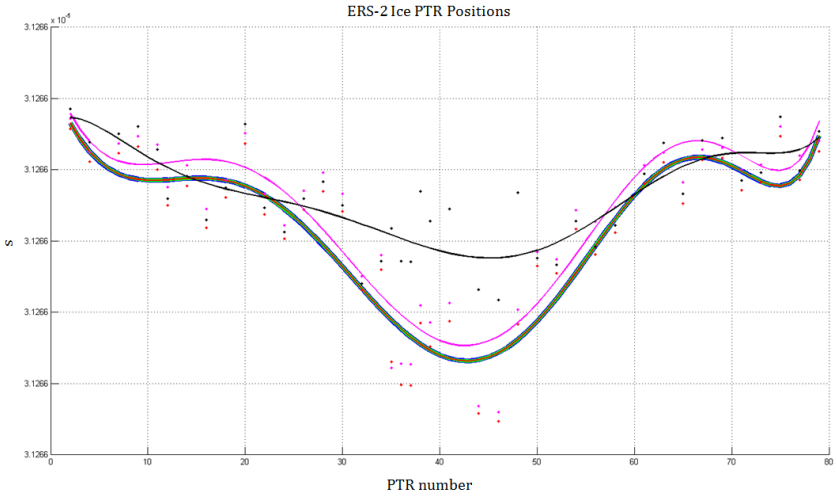


Figure 3.3.25: ERS-2 PTR positions in ice mode together with their trend.

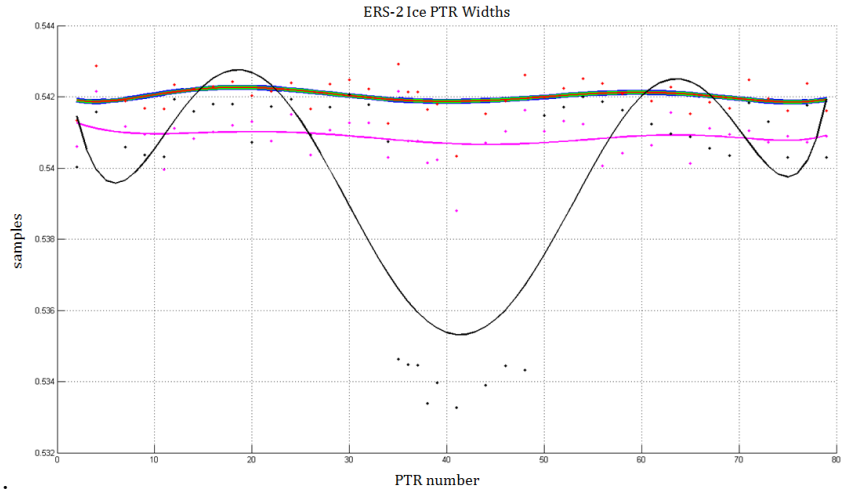


Figure 3.3.26: ERS-2 PTR widths in ice mode and their trend.

- ERAC #2

In this second example, the behaviour of the fitting methods repeats. The 3-Gaussian fitting has an offset with respect to the Three-point and the Simple Gaussian fittings, while the 2-Gaussian fitting follows a totally different trend. Moreover, in this second case, the trend is forced by half of the PTR estimations, located mainly in the first part of the ERAC, which is even larger in the width estimations (Figure 3.3.28) than in the position estimations (Figure 3.3.27).

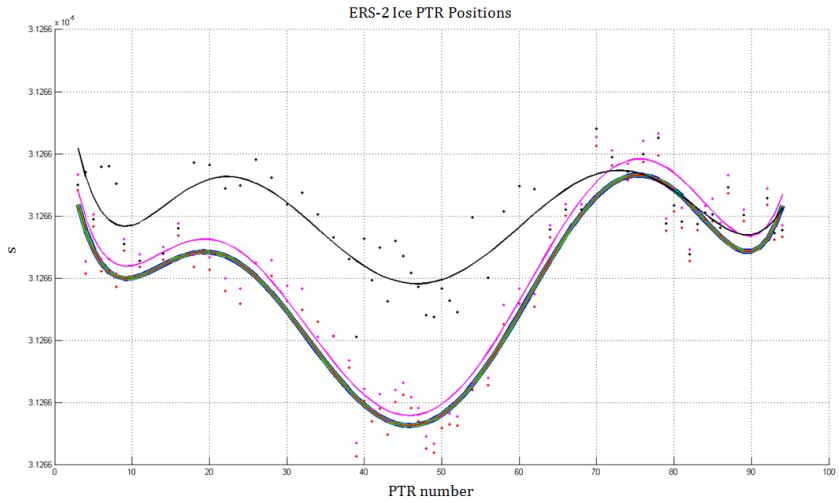


Figure 3.3.27: ERS-2 PTR positions in ice mode and their trend.

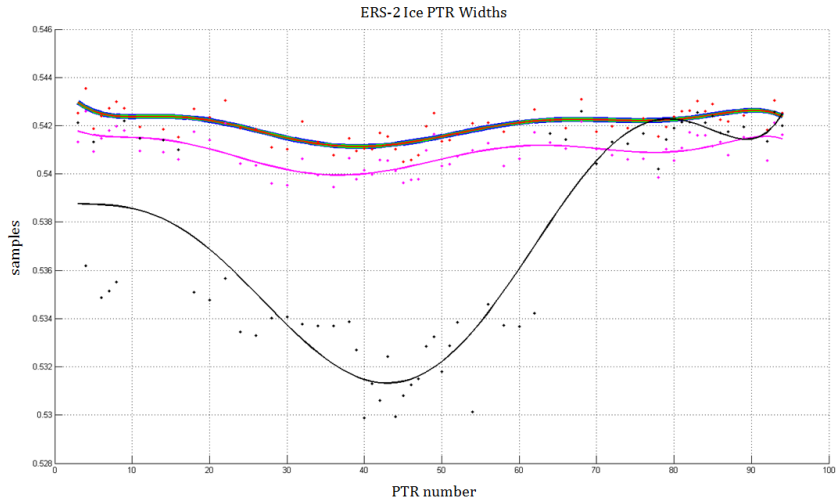


Figure 3.3.28: ERS-2 PTR widths in ice mode and their trend.

3.3.2.4 ERS-2 ocean mode

- ERAC #1

Different from the results in ice mode, the Multiple Gaussian fittings are closer to the other two methods. However, we still see a bias. Apart from that, what really changes from what we saw in ERS-1 is that the Three-point fitting no longer has a bias with respect to the two Simple Gaussian fittings. The PTR positions intersect sometimes (Figure 3.3.29) while the PTR widths follow completely different trends (Figure 3.3.30).

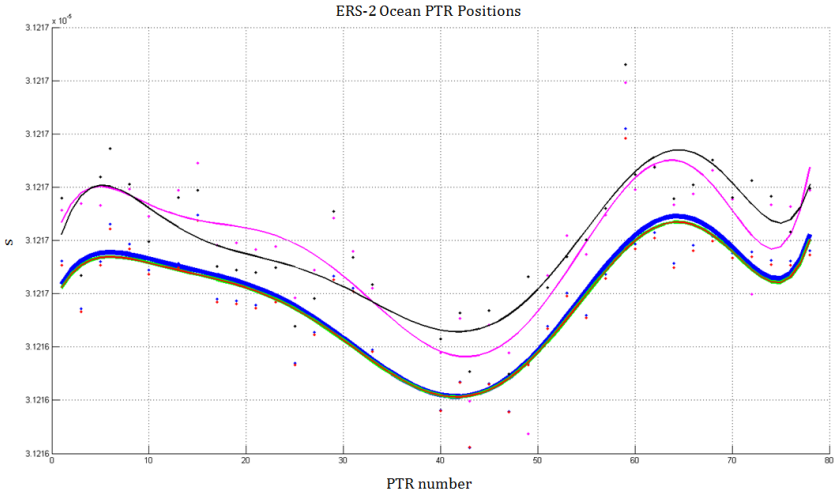


Figure 3.3.29: ERS-2 PTR position in ocean mode and their trend.

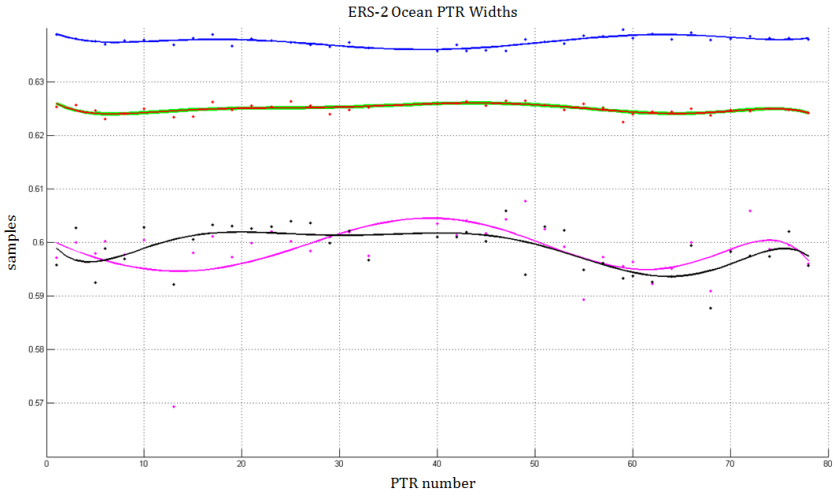


Figure 3.3.30: ERS-2 PTR width for in ocean mode and their trend.

- ERAC #2

In this second example, we see a very similar behaviour, although the 3-Gaussian fitting now differs from the rest of the trends when estimating the PTR position (Figure 3.3.31). And for what regards the PTR width (Figure 3.3.32), the bias between the Three-point fitting and the two Simple Gaussian fittings is also present.

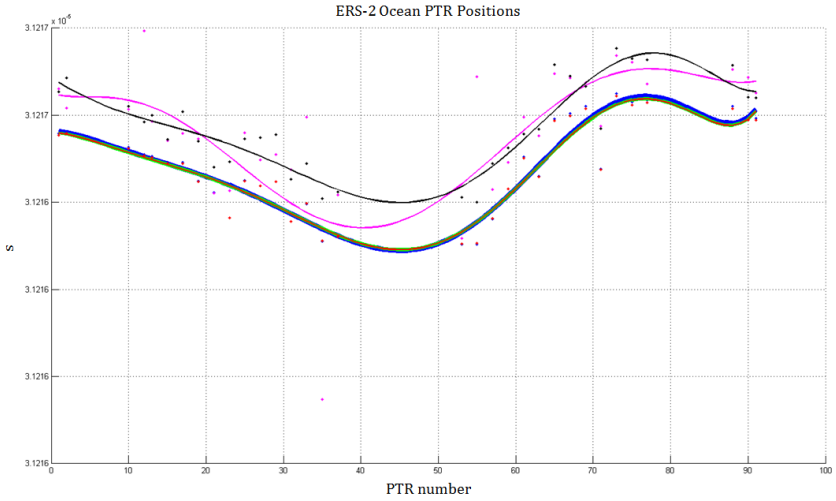


Figure 3.3.31: ERS-2 PTR position in ocean mode and their trend.

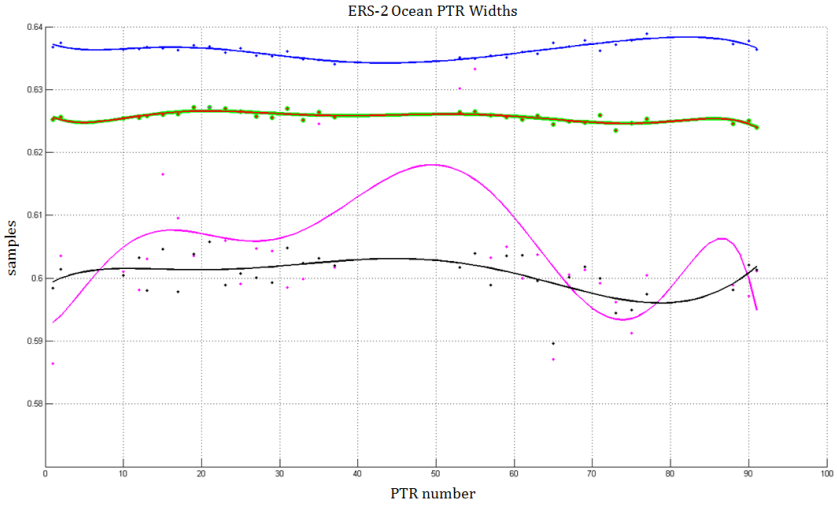


Figure 3.3.32: ERS-2 PTR width in ocean mode and their trend.

3.3.2.5 Results analysis

In Section 3.3.1, we observed that the Multiple Gaussian fittings were the ones that best fitted the PTR samples. However, we noticed that the results highly depend on the seed that is given to the method: only when we provide them with the right seeds, we get a perfect match. When looking at individual PTRs, the method can be tuned to provide a perfect fit using a Multiple Gaussian fitting.

The Simple Gaussian fitting does not fit the samples as well as the Multiple Gaussian. However, the Simple Gaussian fitting is not seed dependent and, specially when we work with a long period of data, the fittings appear to be much more stable. Calculating the right seed for each PTR of a product is not affordable.

Regarding the details of the Mulitple Gaussian fitting, we observe differences between the 2-Gaussian fitting and the 3-Gaussian fitting. The main one is that the 3-Gaussian fitting works well in all cases, whether the PTR is symmetrical or asymmetrical, while the 2-Gaussian fitting only works well when the PTR is asymmetrical (which is the case of ERS-1 but not ERS-2).

Differences between Simple Gaussian and 3-Gaussian fittings in the PTR position estimation, in the cases when the Multiple Gaussian performs the best, are down to roughly 0.3 mm (with secondary lobes) and 0.13 mm (without secondary lobes) in the PTR position estimation.

On the other hand, differences between the Simple Gaussian and the Three-point fittings (the latter being a reference) in the PTR position estimation are of the order of 0.1 mm. We can observe this bias all through the ERAC files we have tested in this phase as a fairly constant value. We should investigate this difference through a longer period of data to confirm it is a bias, a drift or random differences.

For what regards the PTR width, all the above results apply. Estimations of the PTR width with the Simple Gaussian and the Three-point fittings show differences (that look like a bias) that are equivalent to about 0.5 mm. They shall also be investigated within the long-term analysis.

3.3.3 Long-term tests

In this last step of the testing, only the two methods that provided the best results in the previous stages are going to be tested. These two methods are the Three-point fitting and the Simple Gaussian fitting.

The input data for these tests will be one cycle of data for each satellite. In previous tests, we did not pay attention to the overlapping of the two missions when choosing individual ERACs. However, in this last stage we will process cycles that overlap, since they could add some information to our conclusions. For ERS-1 we will use cycle 155 (from 24.03.1996 to 28.04.1996) and for ERS-2 we will use cycle 10 (from 25.03.1996 to 29.04.1996).

We must note that this stage of tests is part of the commisioning phase of the REAPER project. For this phase, an interpolation of the smoothed output data is needed, in order to have one measurement every second. Hence, we will also focus in the smoothing process.

3.3.3.1 Smoothing and interpolation of original data

In all the mid-term tests (Section 3.3.2) we chose to fit the output results of the different methods with a polynomial fitting. After some tests, a 7th degree polynomial fitting was chosen as the most appropriate to fit all the oscillations along the orbit. Thus, we decided to use it for the long-term tests as well.

Once the fitting was performed, in order to interpolate the data, a high sampling of the resulting polynomial was done. When analysing the results, the performance was as good as expected for all the data except for the extremes of each ERAC, where the polynomial was sampled outside the original data boundaries. This effect can be observed in Figure 3.3.33, where the Three-point fitting results were used.

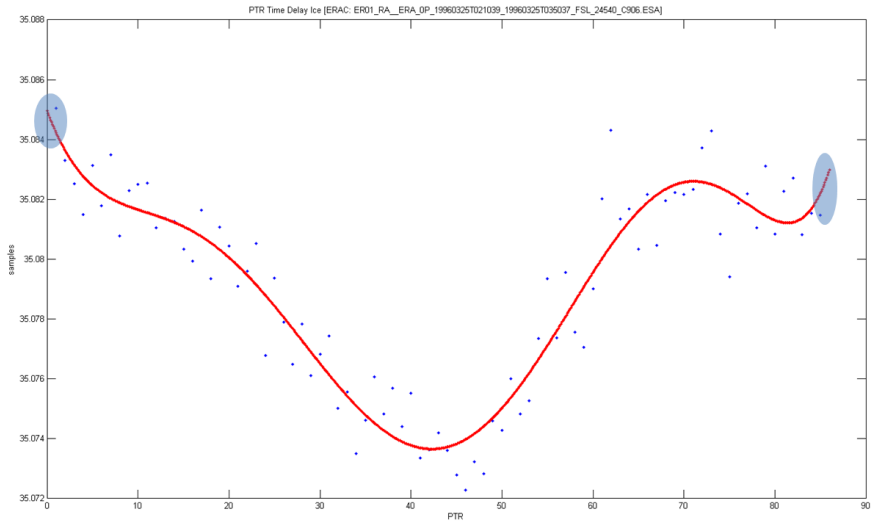


Figure 3.3.33: ERS-1 PTR positions in ice mode and their trend (computed with a 7th degree polynomial fitting).

On the left extreme the first values are extrapolated with the slope of the following values while on the right end the values are extrapolated with the slope of the previous values, giving wrong values.

This effect becomes more obvious when we look at a longer period of time, as shown in Figure 3.3.34.

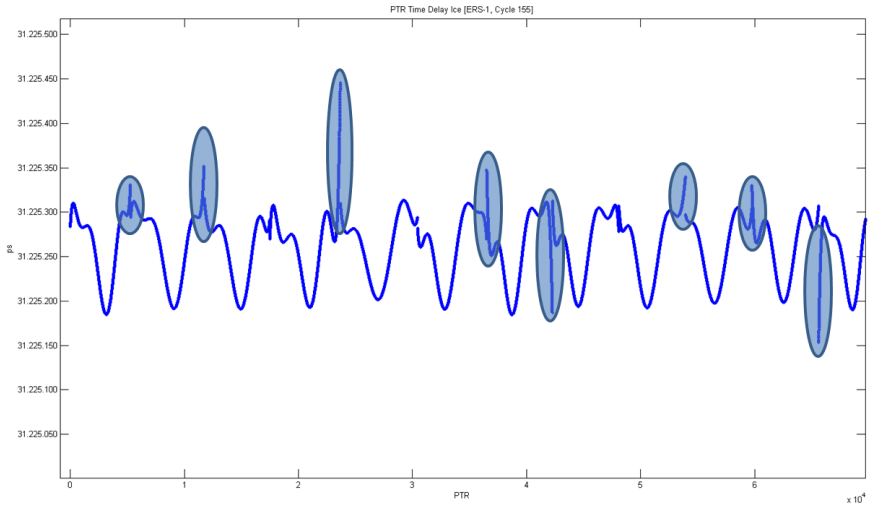


Figure 3.3.34: ERS-1 PTR positions in ice mode and their trend (computed with a 7th degree polynomial fitting). The blue circles point erratic values due to the smoothing and extrapolation.

As we can see, there are a lot of errors at the extremes of every ERAC. Moreover, when there is an anomaly or a switch-off, the results are worse. The following figures show the erratic behaviour in these situations.

Figure 3.3.35 shows an anomaly that introduces some undesired oscillations.

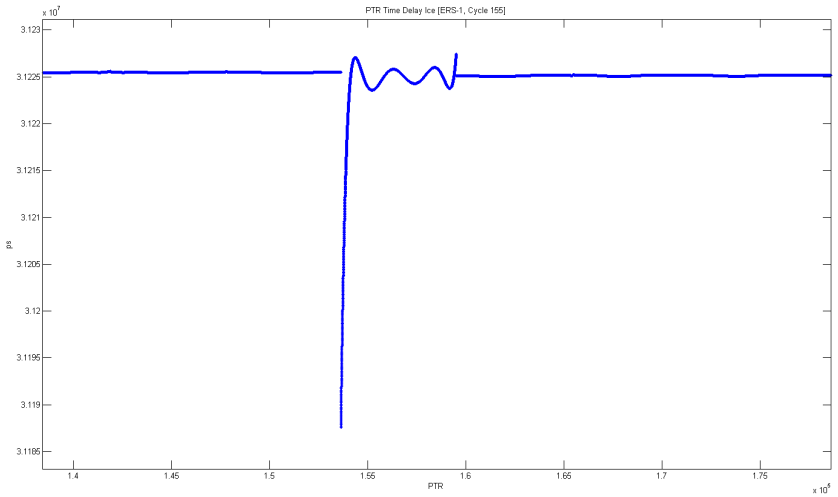


Figure 3.3.35: ERS-1 PTR positions in ice mode and their trend (computed with a 7th degree polynomial fitting).

Now, if we zoom in (Figure 3.3.36), we can observe the effect clearer, seeing that the extra oscillations are much bigger than the orbital oscillations of the PTR time delay.

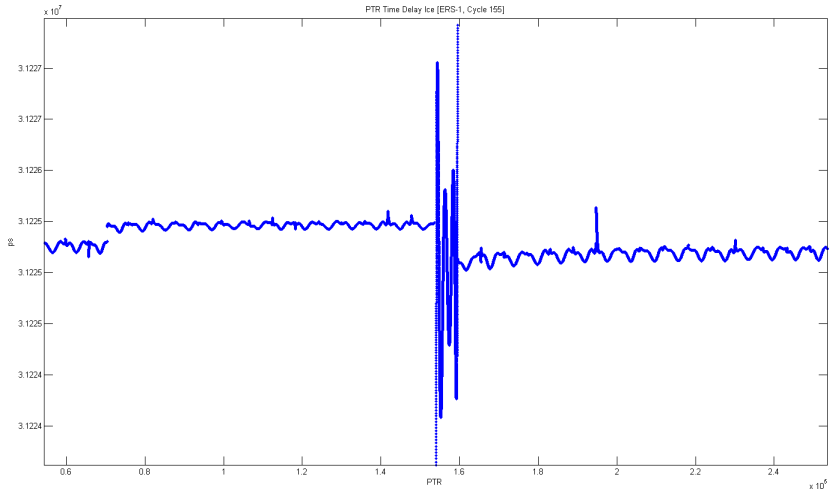


Figure 3.3.36: Zoom of Figure 3.3.35.

Since these results are not acceptable, we decided to change the smoothing function. This time, the smoothing will be done with a running window of 8 samples. After that, a linear interpolation will be performed.

Further, for the particular case of the values that have to be extrapolated, our choice has been to maintain the last valid value (or the first one in the case that the extrapolation is done at the beginning of the ERAC). The results obtained with this second option are the following.

Compared to the first case (see Figure 3.3.33), now we have much better results at the extremes of the ERAC file, as shown in Figure 3.3.37.

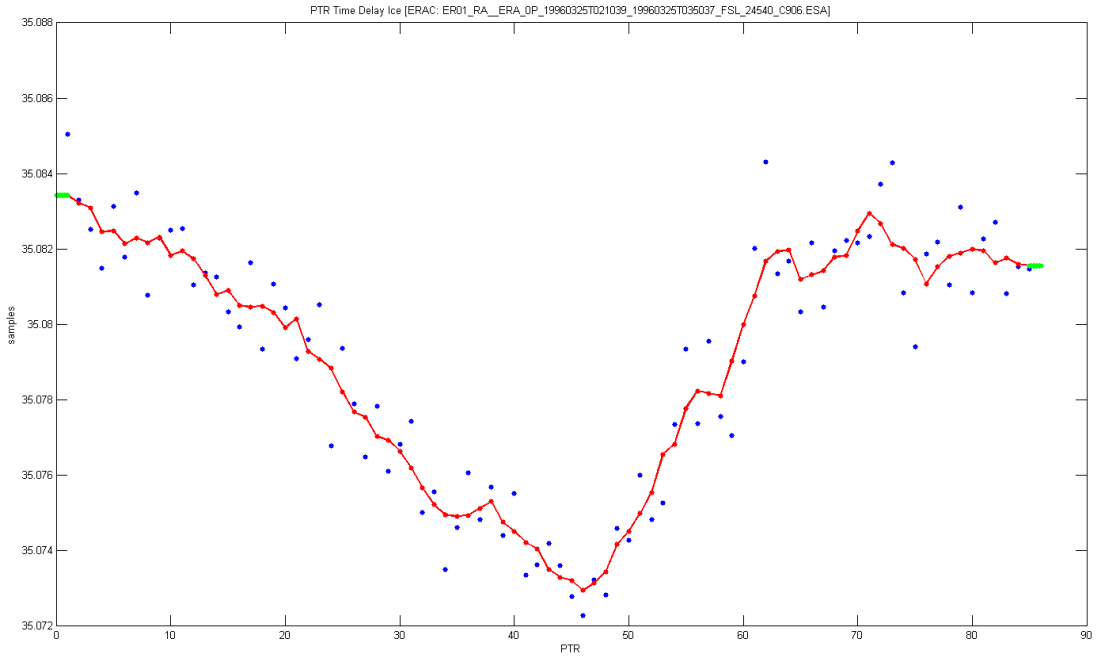


Figure 3.3.37: ERS-1 PTR positions in ice mode. Smoothed with a running window (red) and extrapolated (green).

Figure 3.3.38 and Figure 3.3.39 zoom in at each side of the file.

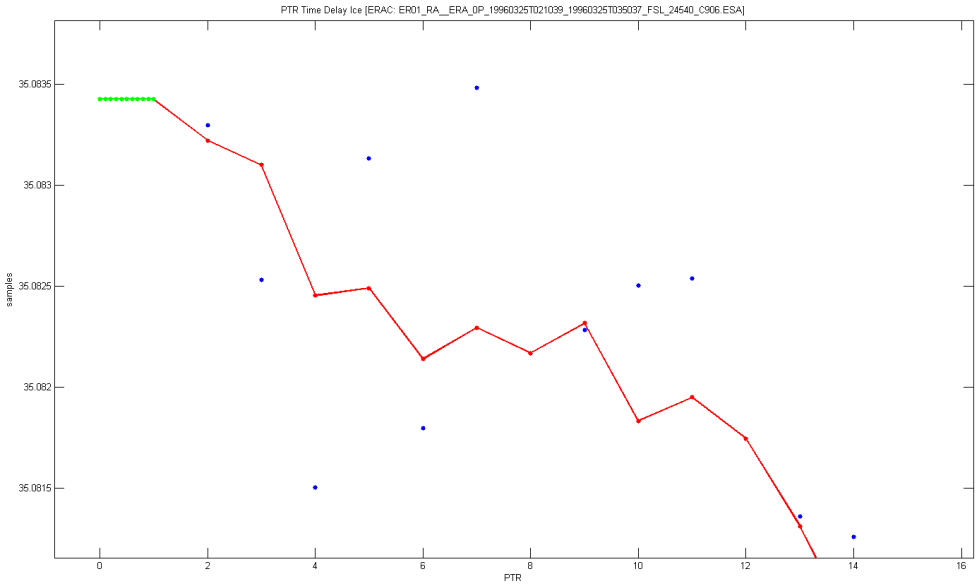


Figure 3.3.38: ERS-1 PTR positions in ice mode. Smoothed with a running window (red) and extrapolated (green).

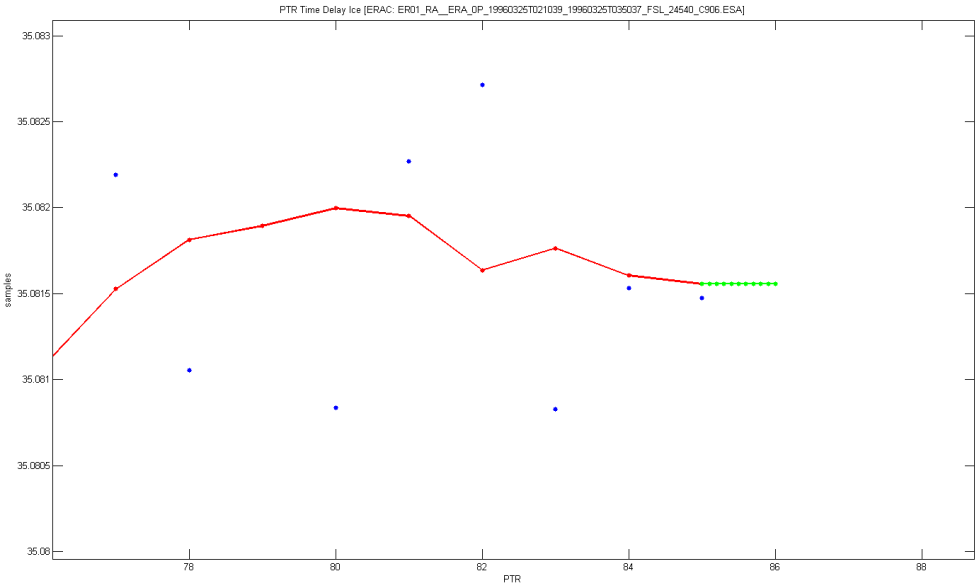


Figure 3.3.39: ERS-1 PTR positions in ice mode. Smoothed with a running window (red) and extrapolated (green).

Further, if we use a longer period as in Figure 3.3.40, we see how results improve.

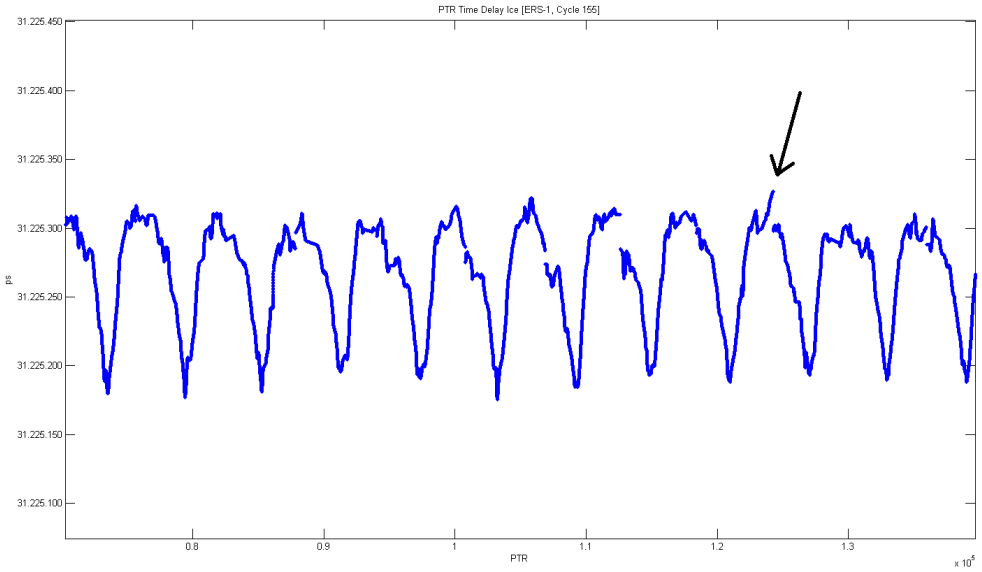


Figure 3.3.40: ERS-1 PTR positions in ice mode and smoothed with a running window. The black arrow points to what may be an anomalous values.

Even that it may seem that we still get some wrong values at the end of one ERAC (Figure), if we zoom in (Figure), we can appreciate that a real value is there and that it has been maintained during the extrapolation. With this, we can say that, at least, the method is not extrapolating with a wrong slope.

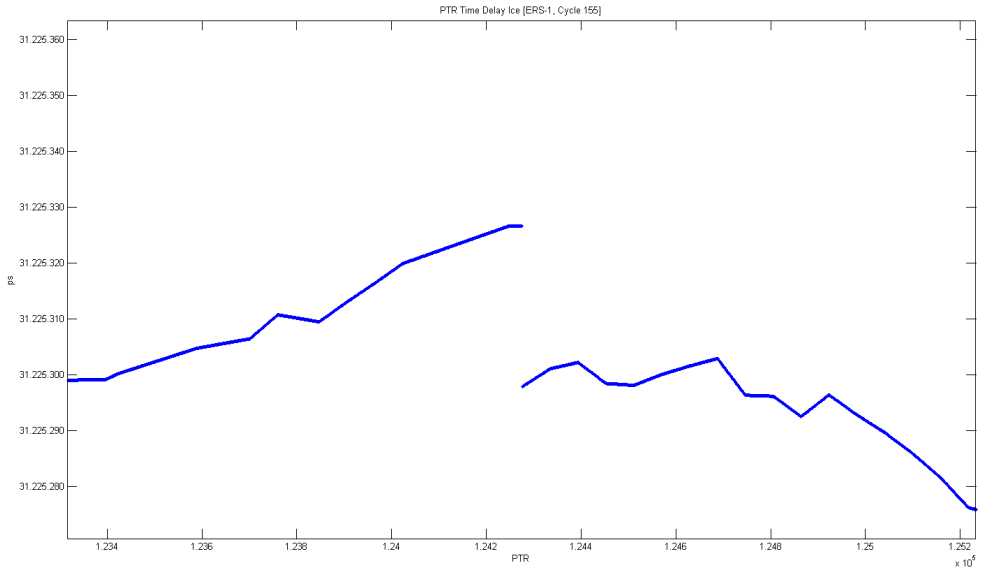


Figure 3.3.41: Zoom of Figure 3.3.40.

Besides, when an anomaly occurs, the smoothing works fine and does not introduce any oscillations. Figure 3.3.42 and Figure 3.3.43 show that.

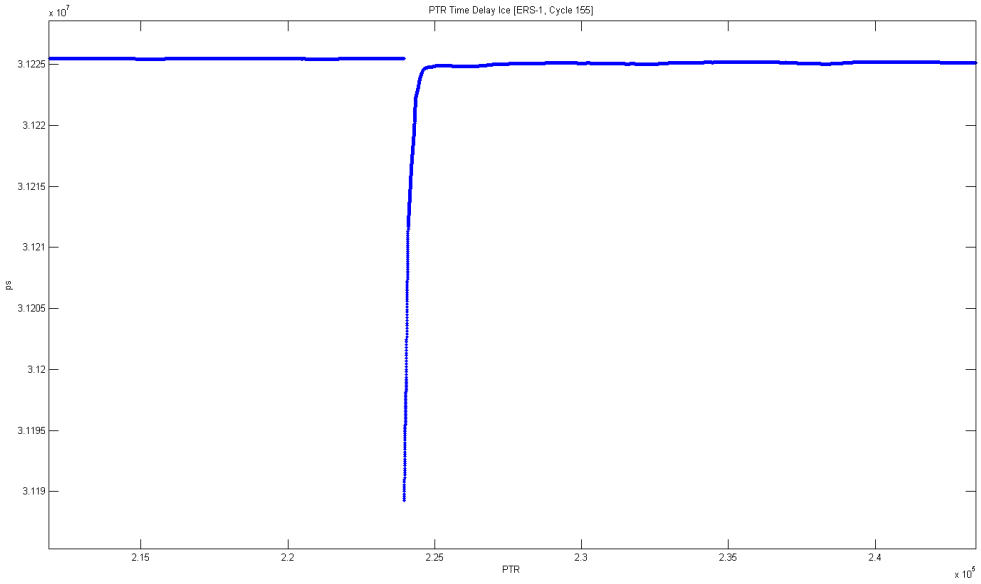


Figure 3.3.42: Switch-off anomaly in ERS-1 PTR positions in ice mode and smoothed with a running window.

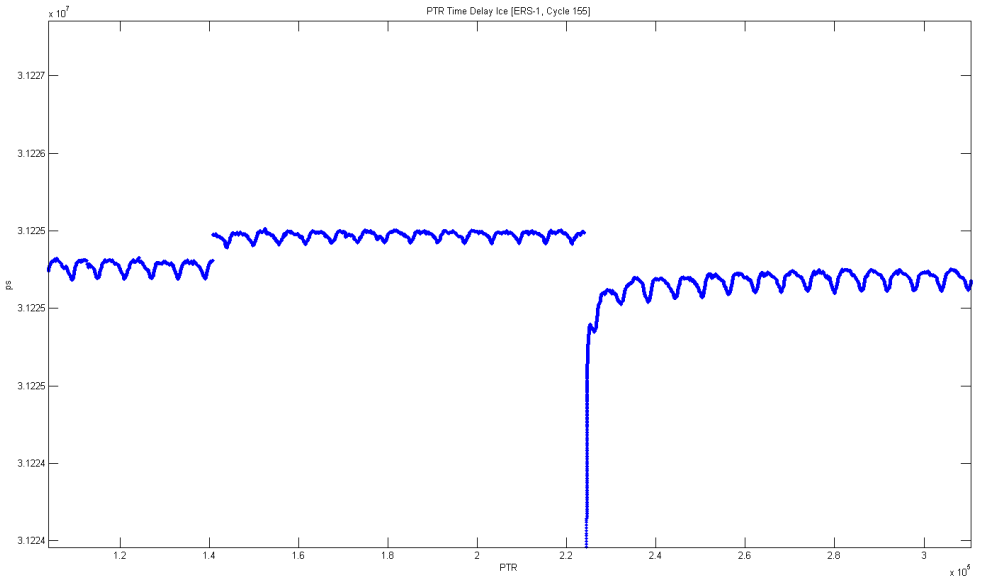


Figure 3.3.43: Zoom of Figure 3.3.42.

With these results, we can conclude that the polynomial fitting is not useful for smoothing the PTR parameters at an ERAC level. Instead, a running window and the linear interpolation that follows will be used.

3.3.3.2 Methods comparison

Now that the smoothing method has been fixed, we can proceed to analyse the results of the long-term tests. Again, for ERS-1 we will process cycle #155 and for ERS-2, cycle #10.

In ERS-1, we will observe some jumps both in time delay and width. These are due to a clock asymmetry that occasioned what was called SPTR jumps after every instrument switch-off and other anomalies. However, these jumps were taken into account and compensated for during the commissioning phase.

The commissioning phase is the stage of a project or mission where a set of engineering techniques and procedures are checked, and every operational component of the project is inspected and tested.

Note: All the following plots share the same legend.

Three-point fitting Simple Gaussian fitting with Th

ERS-1 ice mode In ice mode, the long-term comparison between the two fitting methods for both the ERS-1 PTR time delay (Figure 3.3.44) and PTR width estimations (Figure 3.3.46) shows a complete agreement. Figure 3.3.45 shows that the difference between the position estimations of two methods is 0 everywhere, except for some cases where the differences are of the order of 10^{-20} s, which is completely negligible.

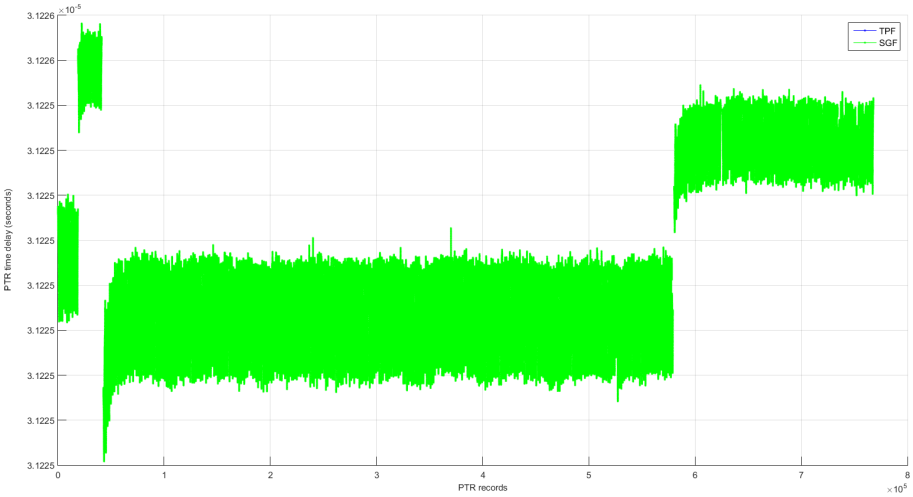


Figure 3.3.44: ERS-1 PTR time delay in ice mode (cycle 155).

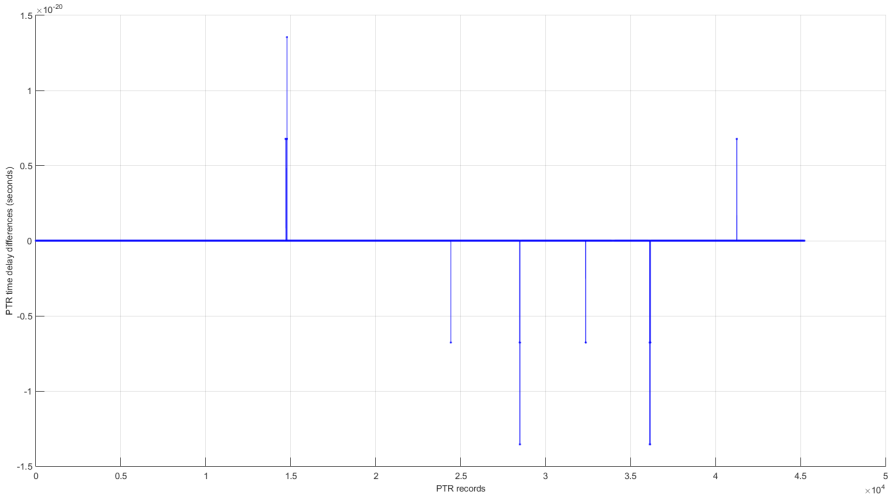


Figure 3.3.45: ERS-1 PTR time delay differences in ice mode (cycle 155).

As far as the comparison between the PTR width estimations is concerned, the outcome is the same. The differences between the two methods are of the order of 10^{-15} samples (see Figure 3.3.47), which is totally negligible.

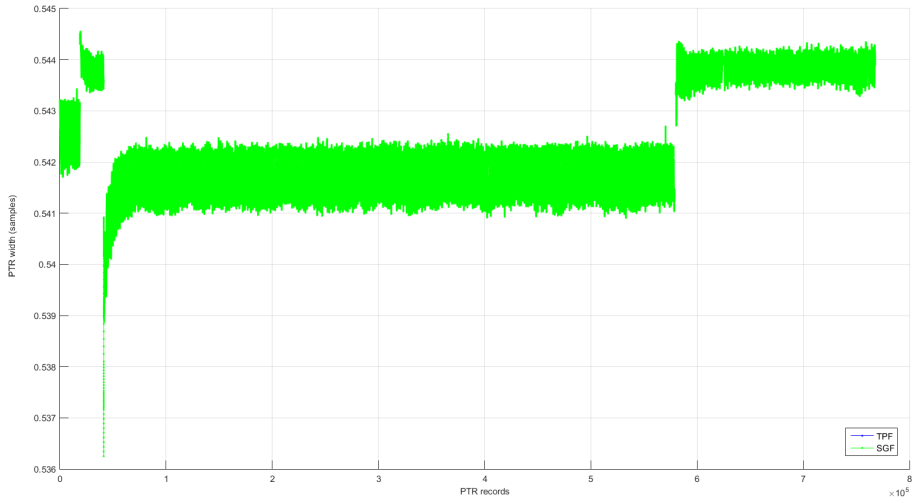


Figure 3.3.46: ERS-1 PTR width in ice mode (cycle 155).

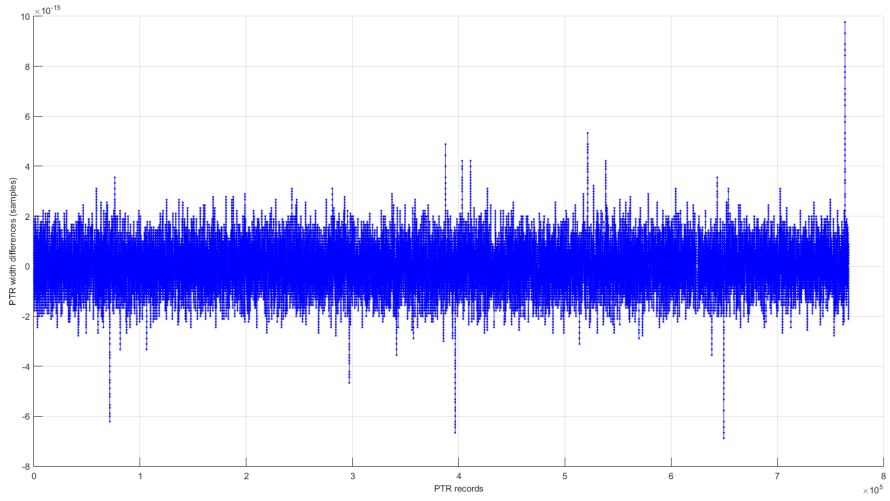


Figure 3.3.47: ERS-1 PTR width differences in ice mode (cycle 155).

ERS-1 ocean mode In the case of ocean mode, the two methods also give very similar outputs (Figure 3.3.48 and Figure 3.3.50), although the differences are bigger. This difference with respect to the ice case is because the Three-point fitting is based on three samples while the Single Gaussian fitting fits all the samples that are above a threshold.

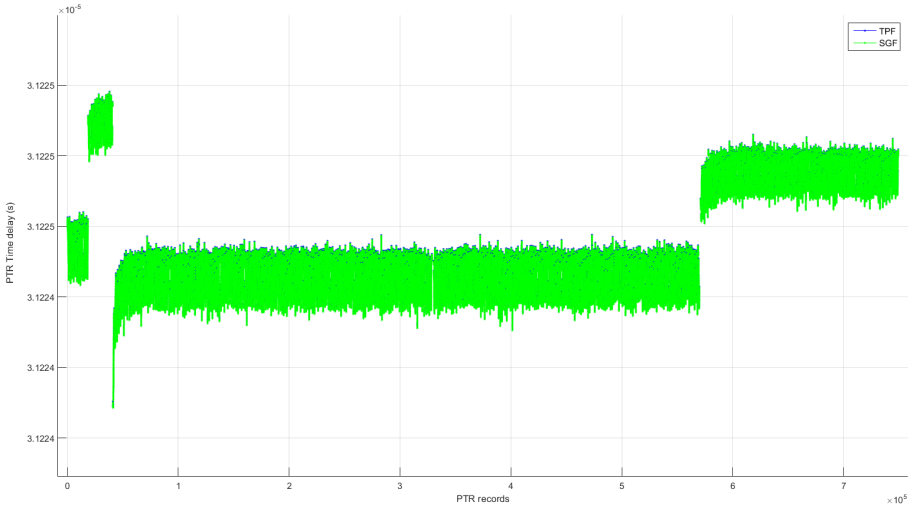


Figure 3.3.48: ERS-1 PTR time delay in ocean mode (cycle 155).

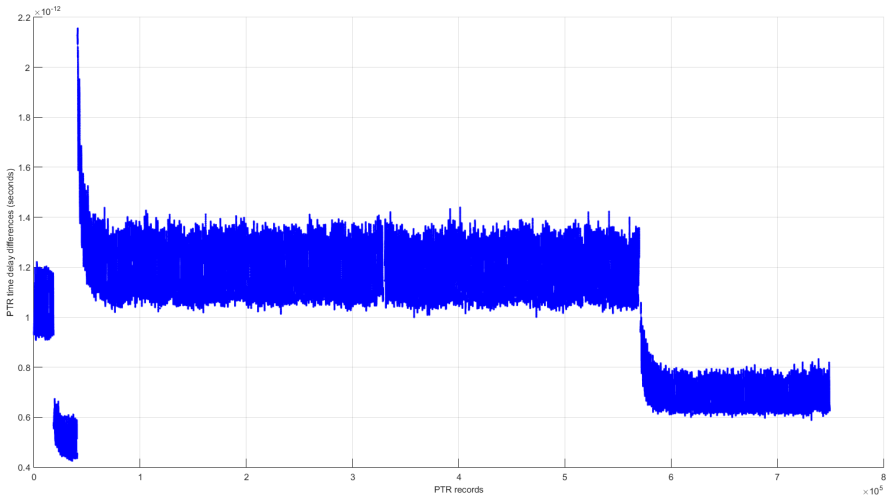


Figure 3.3.49: ERS-1 PTR time delay differences in ocean mode (cycle 155).

This difference is around 1.2 ps, which translates into approximately 0.2 mm in the case of the PTR time delay (Figure 3.3.49) and around 0.001 samples for the PTR width, which in the ocean case means 3 ps (Figure 3.3.51).

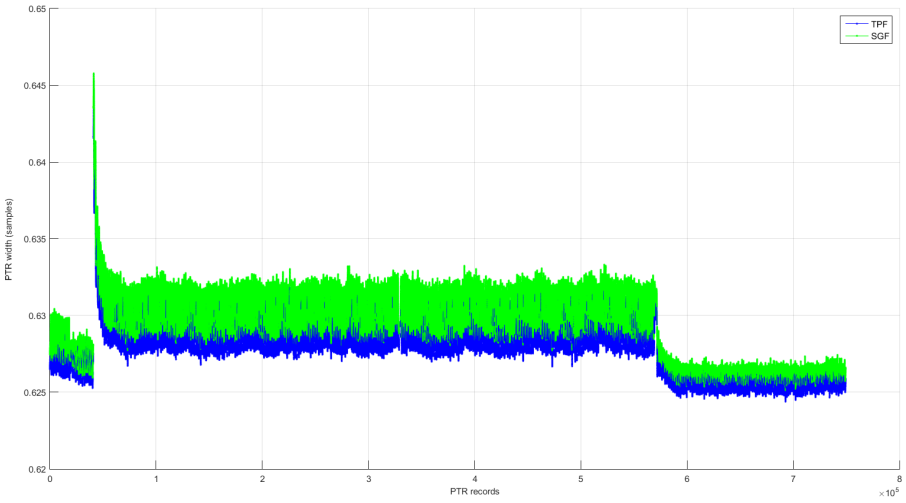


Figure 3.3.50: ERS-1 PTR width in ocean mode (cycle 155).

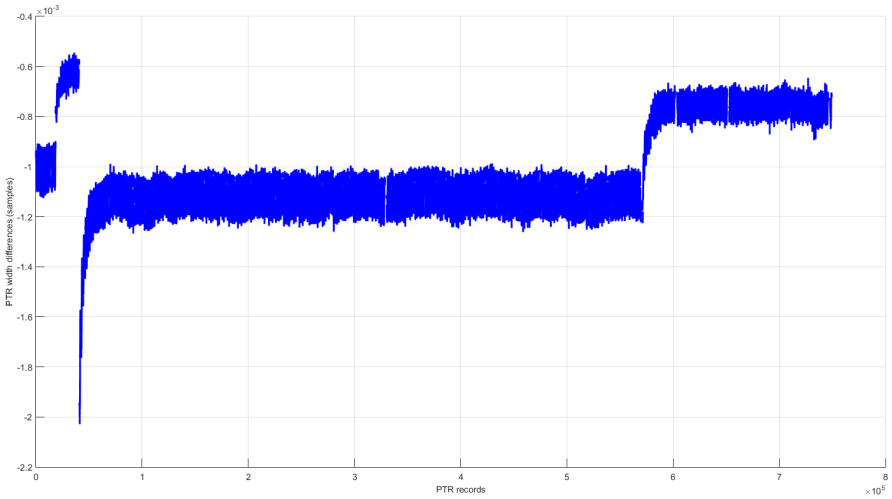


Figure 3.3.51: ERS-1 PTR width differences in ocean mode (cycle 155).

ERS-2 ice mode For ERS-2 in ice mode, we observe the same behaviour as in ERS-1: results from both methods are practically the same. Figure 3.3.52 depicts the PTR time delay estimations while the differences are shown in Figure 3.3.53.

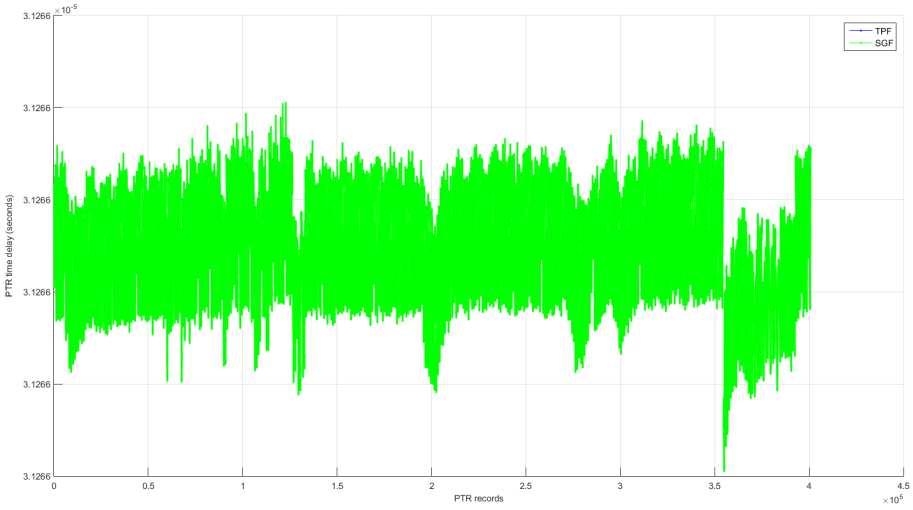


Figure 3.3.52: ERS-2 PTR time delay in ice mode (cycle 10).

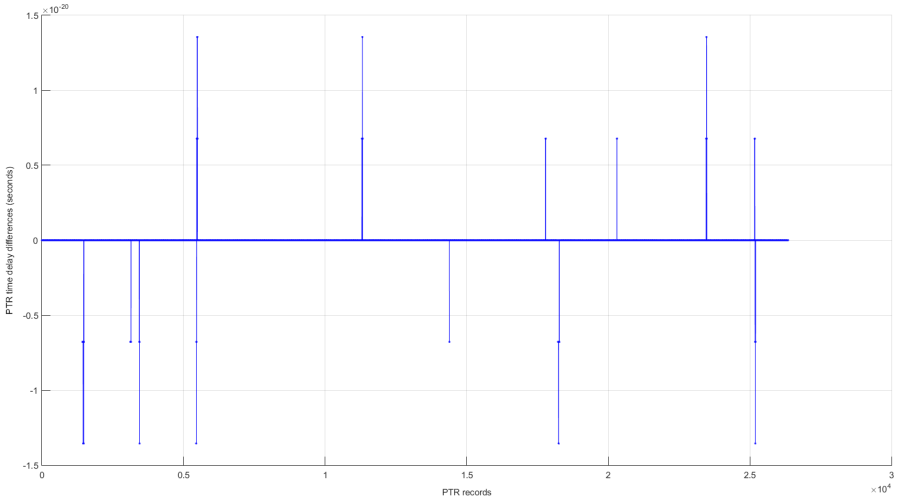


Figure 3.3.53: ERS-2 PTR time delay differences in ice mode (cycle 10).

Figure 3.3.54 shows the PTR width estimations from both modes and Figure 3.3.55 the practically inexistent differences between the two methods.

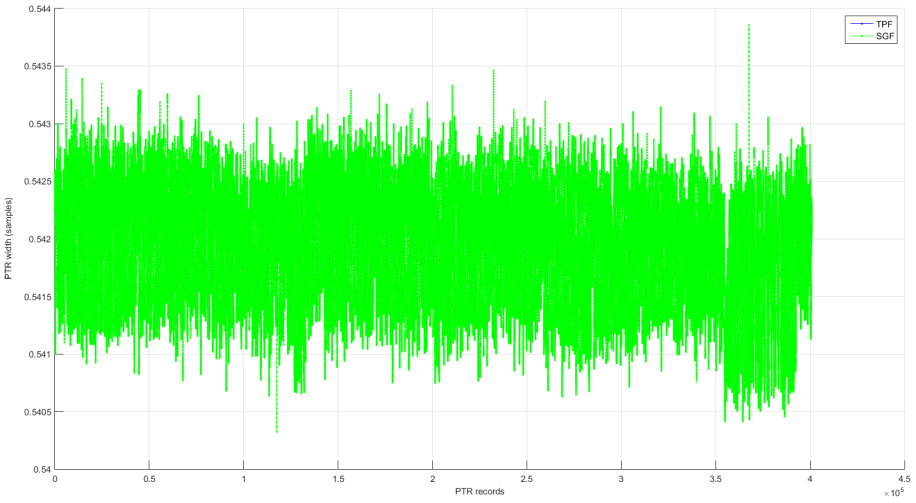


Figure 3.3.54: ERS-2 PTR width in ice mode (cycle 10).

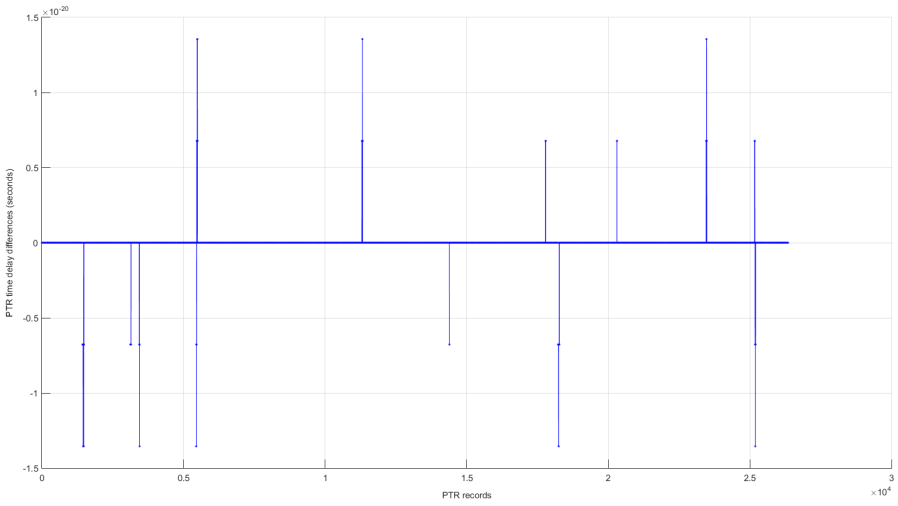


Figure 3.3.55: ERS-2 PTR width differences in ice mode (cycle 10).

ERS-2 ocean mode For the PTR time delay in ocean mode (see Figure 3.3.56), we see that the methods agree as ERS-1, but a bit less. The differences between the estimations from both methods are bigger and up to 1 mm in the case where jumps are present (Figure 3.3.57).

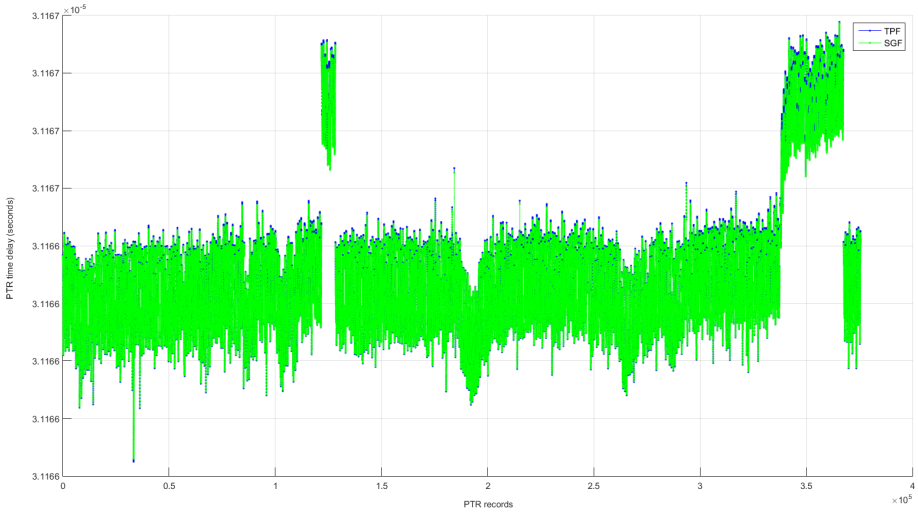


Figure 3.3.56: ERS-2 PTR time delay in ocean mode (cycle 10).

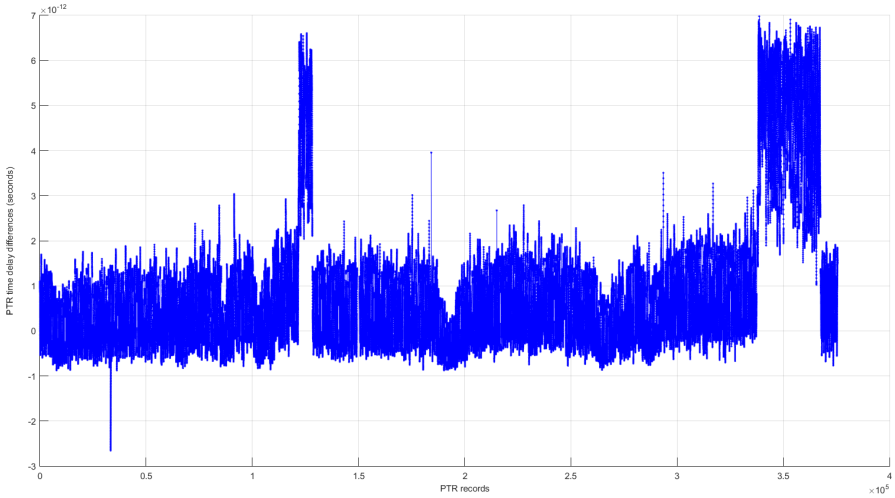


Figure 3.3.57: ERS-2 PTR time delay differences in ocean mode (cycle 10).

However, we do see a bigger difference in the PTR width estimations behaviour. As we had seen in Section 3.3.57, the Three-point fitting was not following the same trend as the Single Gaussian. Figure 3.3.58 shows that, apart from not following the same trend but what seems to be the opposite, there are two big events where the Three-point fitting does not provide values within the expected range.

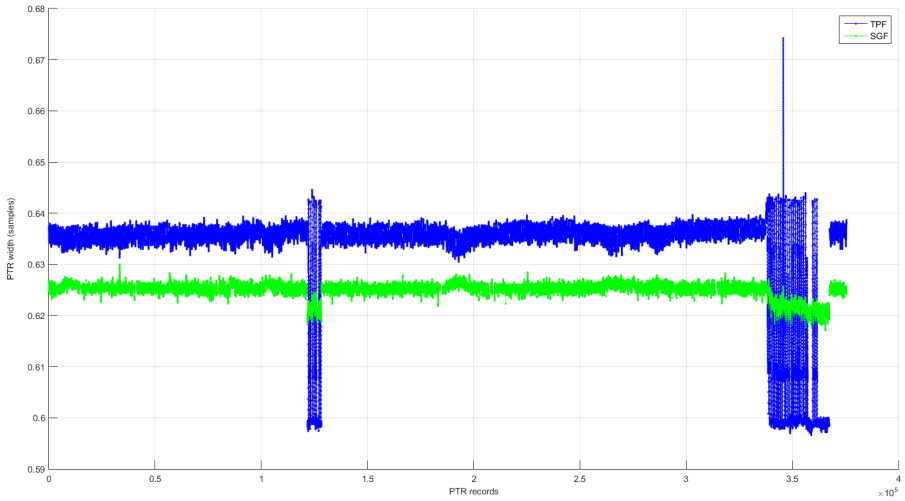


Figure 3.3.58: ERS-2 PTR width in ocean mode (cycle 10).

As we can see in Figure 3.3.59, the differences between both methods are one order of magnitude bigger than the ones observed in ERS-1.

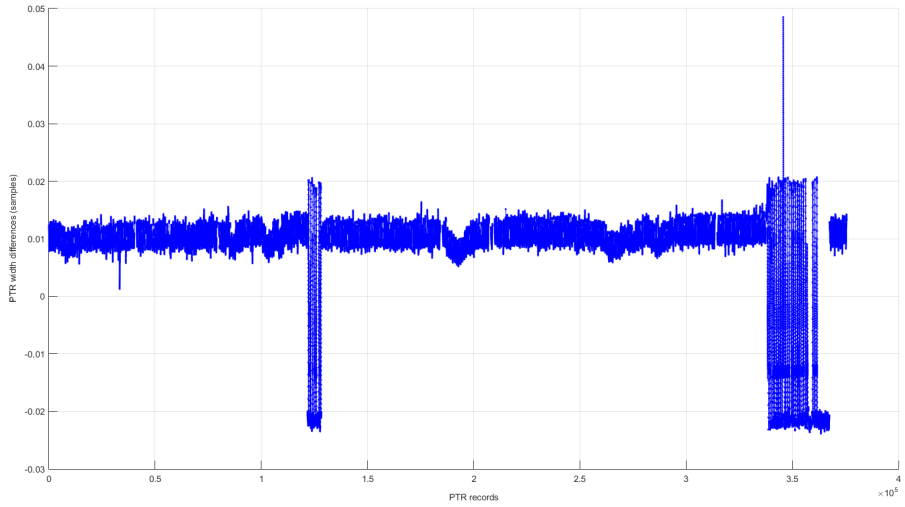
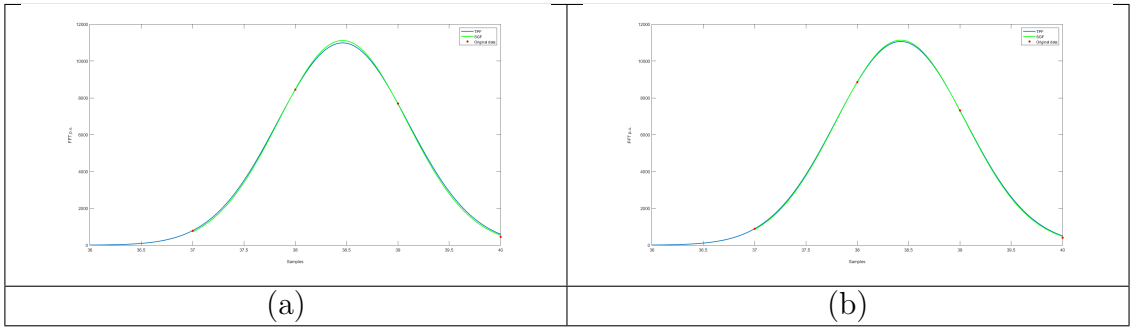


Figure 3.3.59: ERS-2 PTR width differences in ocean mode (cycle 10).

These differences between both methods are due to the shape of the PTR waveforms in ERS-2 for ocean mode. As we mentioned in Section 3.3.2.5, the PTR ocean waveforms in ERS-2 are more or less symmetrical. This means that three

samples (the ones used by the Three-point fitting) cannot represent the shape of a Gaussian as precise as four or five samples would. Thus, the differences between the two modes become bigger. Depending on how symmetric the PTR waveform is, the Three-point fitting outputs change. This can be observed in Figure 3.3.60, where the difference between the two generated Gaussian functions differs more in (a) than in (b) due to the PTR waveform being more symmetrical in the first case.



(a) (b)

Figure 3.3.60: Two symmetrical ERS-2 PTR waveforms (with 4 samples different from 0) in ocean mode (cycle 10), together with both fitting methods.

Results analysis

From the two complete cycles that we have processed we can conclude that when the altimeter is in ice mode, given that the differences between the two outputs are practically inexistent, any of the two methods can be used to estimate the PTR parameters.

As far as the ocean mode is concerned, in ERS-1, although we observed some differences, the Three-point fitting performs as good as the Simple Gaussian fitting. However, when the input signal changes, as in the case of ERS-2 where the PTR waveforms are more or less symmetrical[8], we have seen that the Three-point fitting does not provide as good results as the Simple Gaussian fitting. For this reason, we choose the Simple Gaussian fitting as the best method to estimate the internal path delay.

[8]Note that this phenomenon could also happen in ERS-1 along the mission lifetime due to aging of the instrument, temperature changes, etc.

Chapter 4

Conclusions

After our study was finished, the Simple Gaussian fitting was implemented and integrated in REAPER's operational code to reprocess the whole ERS-1 and ERS-2 missions. These reprocessed data was made public in September 2014.

This new data set, as reported in [17], even though it still had some issues that had to be solved, improved the old missions results. For example, when analysing the standard deviations of crossover height differences (Figure 4.0.1), REAPER data for ERS-1 performed better than the historical reference data. On the other hand, ERS-2 results did not improve that much, but because the original ERS-2 data was of better quality than that of ERS-1. However, during the entire mission REAPER data outperformed the old data set and neared those of Envisat.

Apart from the range, the L2 validation report [17] also evaluated the significant wave height, finding that REAPER data had lower standard deviation with respect to both the old data set and the TOPEX mission (as shown in Figure 4.0.2) meaning that the accuracy was improved.

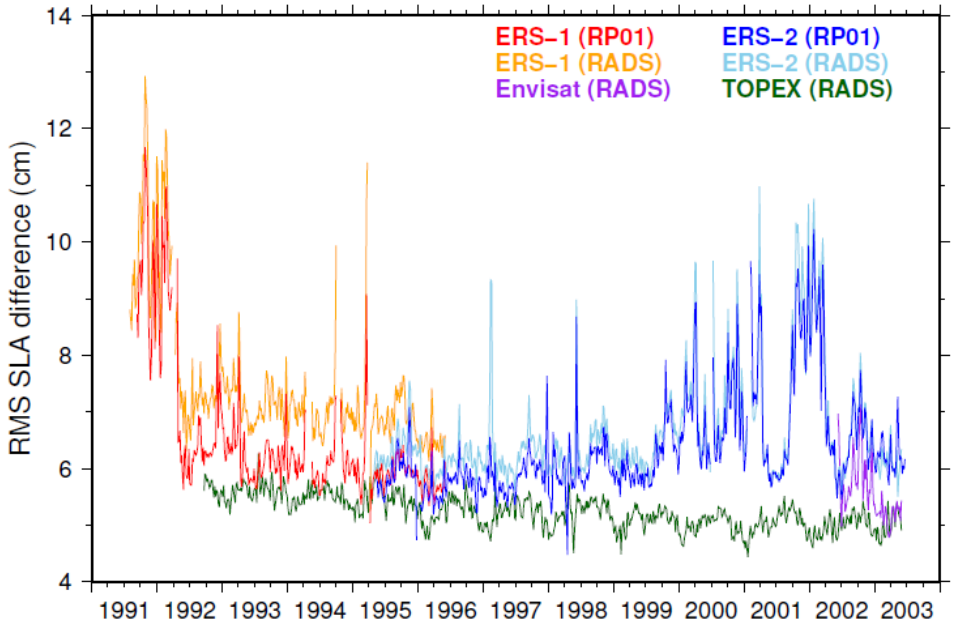


Figure 4.0.1: Time series of RMS single-satellite crossover differences of sea level anomaly. Every value spans all crossovers within a 5-day time span, having passes crossing within 5 days.

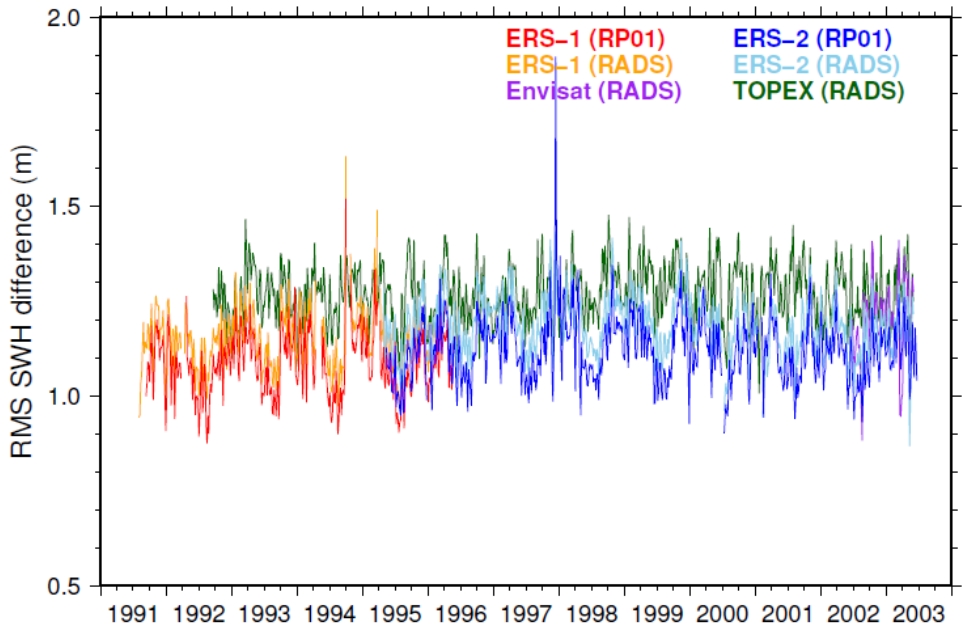


Figure 4.0.2: Time series of RMS single-satellite crossover differences of significant wave height. Every value spans all crossovers within a 5-day time span, having passes crossing within 5 days.

The outcome of this report is in line with other studies on altimeter calibrations. For example, within the CCI: Sea Level project, some corrections in the calibration chain represented a huge change in Envisat mission results on SSH [18].

As this report states, a correction on the PTR internal path delay had a direct impact on the mean sea level trend of more than 2 mm/year, as depicted in Figure 4.0.3.

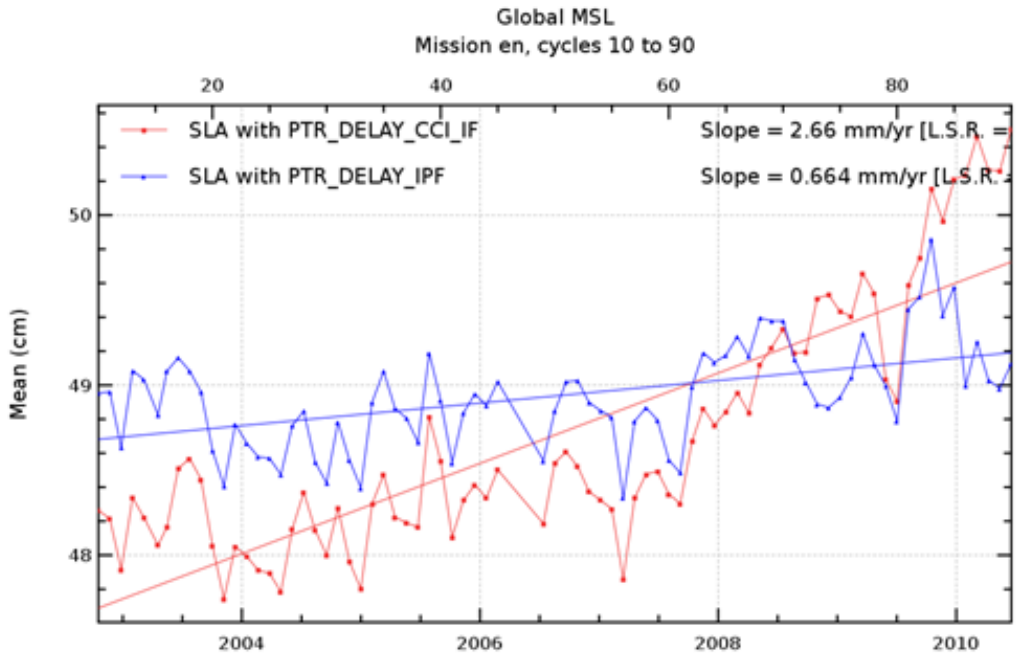


Figure 4.0.3: Global MSL trend comparison.

These changes, even it may seem that a few millimeters are of no importance, since the global mean sea level trend is around 3 mm/year (see [5, 19]), are of huge impact. To conclude, Figures 4.0.4 and 4.0.5 show the Global Mean Sea Level trend before and after applying the mentioned corrections on Envisat, respectively.

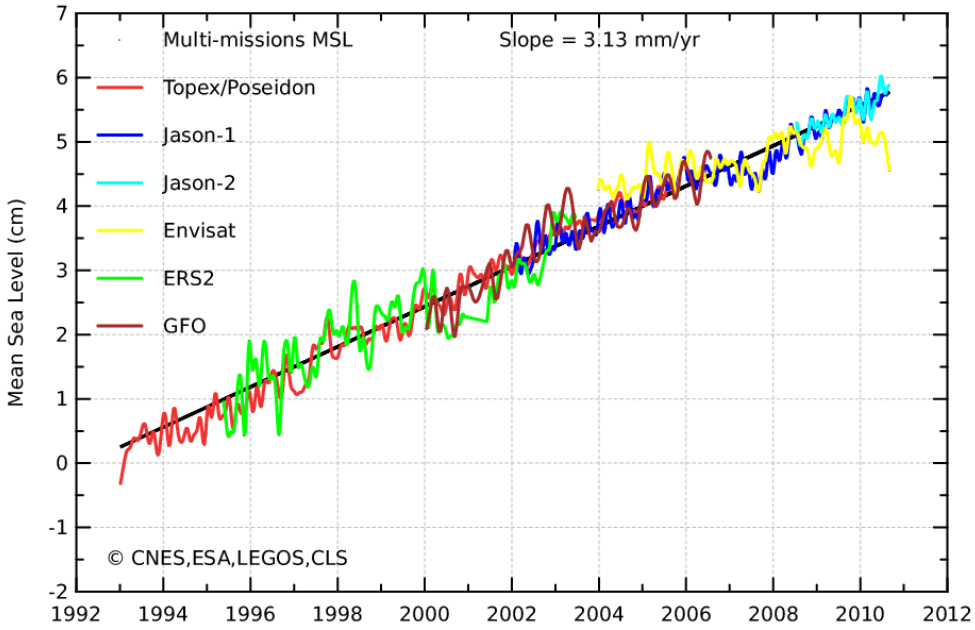


Figure 4.0.4: Multi-mission MSL trend comparison (before corrections on Envisat).

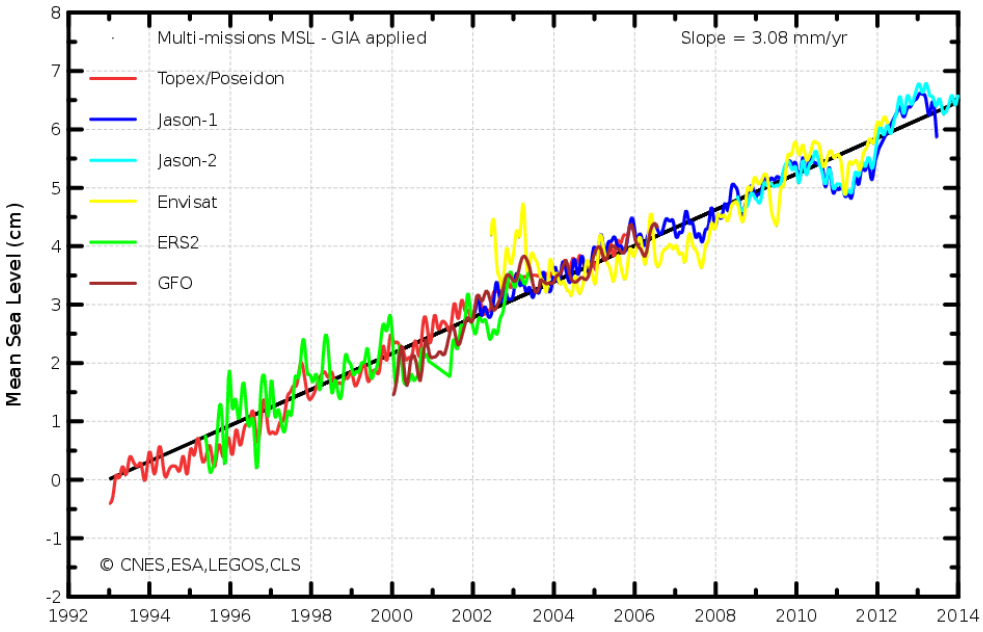


Figure 4.0.5: Multi-mission MSL trend comparison (after corrections on Envisat).

Chapter 5

Annex: Data reading, extraction and selection

Another objective of this project was to gather all the documentation referring to the satellite data structure and its unpacking and decoding, review it and make it precise and clear.

In order to ease the understanding of how the algorithms work, we will divide this chapter in three sections: the definition of the input data, its extraction, and the selection of this data in order to identify the different types of data stored in the products, corresponding to the different instrument working modes.

5.1 Input data

The main input data to our study is the ERAC file. The ERAC file is practically the raw product from the satellite radar altimeter. It is also called L0 product.

In order to read and extract the data from the ERAC, we had to review ERS documentation, which was very old. Apart from that, some specifications were ambiguous or, in very particular cases, inexistent. Hence, our first goal was to be concise and to complete the specifications.

Definition of the input data

The first step was to get the data from the ERAC files and thus the ERAC's structure had to be clarified. The specifications stated that the ERAC was composed of:

- Main Product Header (MPH), where general information about the product is stored,
- Specific Product Header (SPH), which not concerns REAPER, and
- Source Packets (SP), that at the same time, contain:
  - Primary Header, that contains information about the working mode,
  - Secondary Header, with the datation data,
  - Auxiliary Data Block, which stores data used to understand science data,
  - 20 Science Data Blocks, that contain measured scientific data as, for example, the waveform samples.

Initially, this structure made sense as all the actual L0 products have a similar structure (some headers at the beginning, followed by the scientific data). But after the first extraction, we realised the results that we were having were nonsense and the specifications were wrong. The issue was that the MPH and the SPH were actually included in the Source Packets[1]. Thus the specification of the ERAC's structure had to be modified. Figure 5.1.1 shows how it changed.

[1]However, these fields were empty except for the first source packet.

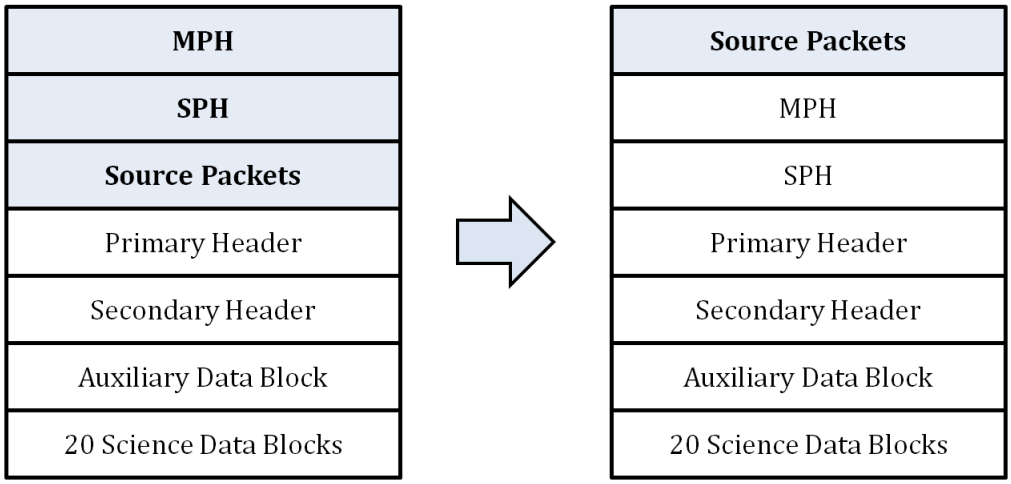


Figure 5.1.1: The ERAC's structure was wrongly specified.

Then, with the structure defined, we were ready to begin the unpacking of the data. Next tables show all the fields contained in the Source Packets (except for the MPH, where the most relevant fields are showed, and for the SPH, which will not be described for the reasons mentioned above).

MPH (176 bytes)				
Bytes	**Parameter Name**	**Code name**	**Type**	**Units**
0-16	Product identifier	product_id	17 bytes	Various
18	Spacecraft identifier	spacecraft_id	1 byte	None
19-42	UTC time of subsat point at the beginning	utc_time_subsat0	24 bytes	Date - Time
43	Station identifier	station_id	1 byte	None
46-69	UTC time when MPH was generated	utc_time_gen	24 bytes	Date - Time
84-107	UTC reference time	utc_ref_time	24 bytes	Date - Time
108-111	Reference binary time of satellite clock	ref_bin_time	4 bytes	Count
112-115	Step length of satellite clock	step_length	4 bytes	ns
128-151	UTC time of ascending node state vector	utc_time_asc	24 bytes	Date - Time

Table 5.1.1: Main fields of the MPH.

Primary Header (6 bytes)				
Bytes	**Parameter Name**	**Code name**	**Type**	**Units**
0-1	Packet identifier	ra_status_id	2 bytes	Bit field
2-3	Packet sequence control data	ra_seq_cntrl	2 bytes	Count
4-5	Packet length indicator	pkt_len	2 bytes	Count

Table 5.1.2: Primary Header.

Secondary Header (6 bytes)				
Bytes	**Parameter Name**	**Code name**	**Type**	**Units**
6-10	S/C binary counter	sc_bin_count	5 bytes	Clock units
11	Data subset counter	Not used	1 byte	Count

Table 5.1.3: Secondary Header.

Auxiliary Data Block (44 bytes)				
Bytes	**Parameter Name**	**Code name**	**Type**	**Units**
12-13	α-HTL filter mantissa	a_htl_coeff	2 bytes	None
14-15	α-HTL filter exponent	a_htl_coeff	2 bytes	None
16-17	β-HTL filter mantissa	b_htl_coeff	2 bytes	None
18-19	β-HTL filter exponent	b_htl_coeff	2 bytes	None
20-21	α-STL filter mantissa	a_stl_coeff	2 bytes	None
22-23	α-STL filter exponent	a_stl_coeff	2 bytes	None
24-25	β-STL filter mantissa	b_stl_coeff	2 bytes	None
26-27	β-STL filter exponent	b_stl_coeff	2 bytes	None
28-29	α-AGC filter mantissa	a_agc_coeff	2 bytes	None
30-31	α-AGC filter exponent	a_agc_coeff	2 bytes	None
32-33	β-AGC filter mantissa	b_agc_coeff	2 bytes	None
34-35	β-AGC filter exponent	b_agc_coeff	2 bytes	None
36-37	Power reference value	p_ref	2 bytes	FFT p.u.
38-43	Spare	-	6 bytes	-
44-45	Preset duration	pre_duration	2 bytes	Base frames
46-47	Preset time delay coarse word	pre_t_delay	2 bytes	12.5 ns
48-49	Preset time delay fine word	pre_t_delay	2 bytes	12.5 ns
50-51	Preset time delay rate	pre_t_delay_rate	2 bytes	12.5 ns/PRI
52-53	Preset AGC	pre_agc	2 bytes	dB
54-55	Preset slope	pre_slp	2 bytes	Slope units
56-547	Receiver offset	rx_offset	2 bytes	12.5 ns
58-91	Spare	-	34 bytes	-

Table 5.1.4: Auxiliary Data Block.

Science Data Blocks (repeated for each Data Block)				
Bytes	**Parameter Name**	**Code name**	**Type**	**Units**
n+0	Mode identifier	HRB_mode_id	2 bytes	None
n+2	Noise floor estimation	noise_floor	2 bytes	FFT p.u.
n+4	HTL discriminator output	htl_discr_out	2 bytes	12.5 ns
n+6	STL discriminator output mantissa	stl_discr_out	2 bytes	Slope units
n+8	STL discriminator output exponent	stl_discr_out	2 bytes	Slope units
n+10	AGC discriminator output	agc_discr_out	2 bytes	FFT p.u.
n+12	HTL beta branch	htl_beta_branch	4 bytes	12.5 ns/PRI
n+16	FFT samples (x64)	wfm_bin	64 x 2 bytes	FFT p.u.
n+144	Time delay coarse word	t_delay	2 bytes	12.5 ns
n+146	Time delay fine word	t_delay	2 bytes	12.5 ns
n+148	Slope	slope	2 bytes	Slope units
n+150	AGC	agc	2 bytes	dB

where n = 92, 244, 396, ..., 2980.

Table 5.1.5: Science Data Blocks.

Further, as far as the Science Data Blocks are concerned, there has to be a re-alignment of some parameters. This means that some parameters in the Data Block belong to the previous one and must be re-ordered. This happens because each Data Block represents a 50-pulse average waveform and the FFT is not performed until the buffer is filled. As a consequence, the FFT samples are stored once its corresponding Data Block has been written. Hence, the FFT samples are written in the next Data Block. Thus, some parameters derived from the FFT process also need to be re-aligned. This parameters are the noise floor estimation and all the *disciminator* values. Moreover, because of this re-alignment, the last Data Block from each ERAC must be discarded as it can not be filled with the first Data Block of the next ERAC file.

5.2 Unpacking and Decoding

In this section we will describe the way each parameter has to be extracted from the ERAC file and which transformations have to be applied. Further, we will include the conversion to engineering units of all the variables.

At this point, is essential to define the byte organisation, which will be valid for the rest of the document hereafter, unless otherwise specified. REAPER's byte organisation is different from what may be called as standard. While the Most and Least Significant bits remain the same (the first and the last bit from the left, respectively), the bit numbering within the byte is reversed, being bit 7 the LSb and bit 0 the MSb as shown in Figure 5.2.1.

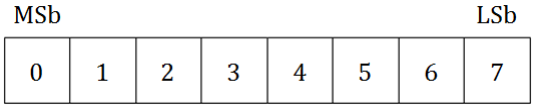


Figure 5.2.1: Byte organisation.

Primary Header

Packet identifier The packet identifier (from now on, packet ID) is a binary number that has information about the Source Packet instrument modes. This information is contained in bits 5-15. Table 5.2.1 shows the meaning of each bit.

Definition	Bit	Settings
Instrument ID bit 1	5	Off
Instrument ID bit 2	6	On
Instrument ID bit 3	7	On
Tracking on Ocean	8	Off = No / On = Yes
Tracking on Ice	9	Off = No / On = Yes
Acquisition on Ocean	10	Off = No / On = Yes
Acquisition on Ice	11	Off = No / On = Yes
BITE mode (Built In Test Equipment)	12	Off = No / On = Yes
Closed Loop Calibration	13	Off = No / On = Yes
Ground test mode (Remote Signal Simulator, RSS)	14	Off = No / On = Yes
Ground calibration mode	15	Off = No / On = Yes

Table 5.2.1: Bits of the packet ID.

Among all modes, the ones that will be interesting for us in Chapter 3 are the Tracking on Ocean and Tracking on Ice modes. During these modes, the satellite takes scientific measurements while during acquisition mode the satellite tries to retrieve the tracking when it has been lost.

Packet sequence control data The packet sequence control data, or source sequence count, is a module 2^{14}counter, incremented by each Source Packet generation and reset to zero after each instrument mode change. The information of this counter is in its last 14 bits as the the figure below shows.

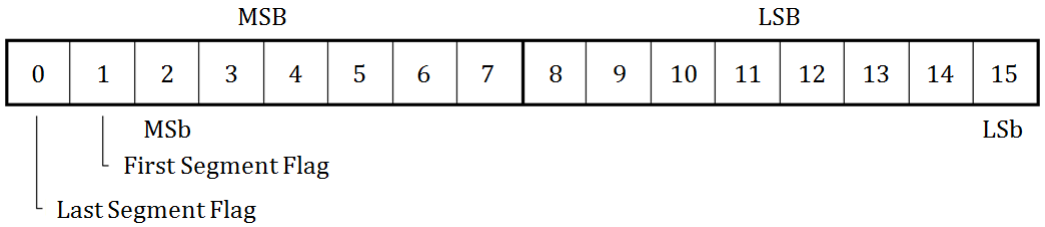


Figure 5.2.2: Source sequence count's structure.

Packet length The packet length refers to the length in bytes of the remainder of the data structure. This number is fully contained in its corresponding two bytes.

Secondary Header

S/C binary counter The S/C binary counter extends from byte 6 (MSB) to 10 (LSB) but with the following distribution for byte 10.

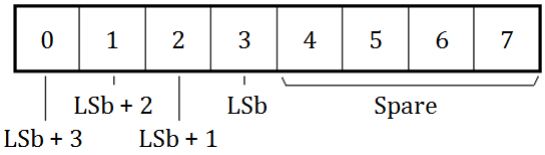


Figure 5.2.3: S/C binary counter's structure.

Auxiliary Data Block

In this block, some variables are extracted in a different way depending on the instrument mode which, for this block, will be Tracking on Ocean or Tracking on Ice. Thus, there will be a selection of data based on whether the bit 8 or 9 from the packet ID are activated.

α-HTL filter coefficient This parameter is held as a 2's complement mantissa/exponent pair and its format is:

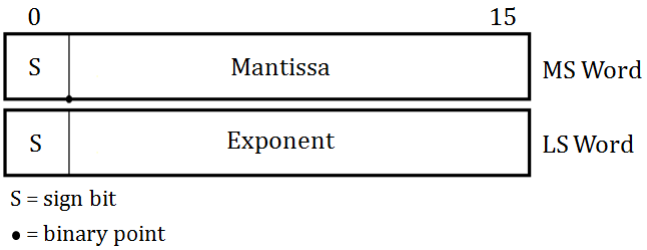


Figure 5.2.4: α-HTL filter coefficient's format.

The mantissa/exponent pair must be extracted the following way:

$$telemetred\_value = mantissa * 2^{exponent}, \tag{5.2.1}$$

and converted to engineering units with the expression:

- Ocean mode

$$\alpha\_htl\_coeff = \frac{telemetred\_value * 1e3}{time\_factor\_ocean/\left[25 * (p\_ref/100)\right]} \tag{5.2.2}$$

- Ice mode

$$\alpha\_htl\_coeff = \frac{telemetred\_value * 1e3}{time\_factor\_ice/\left[12.5 * (p\_ref/100)\right]} \tag{5.2.3}$$

β-HTL filter coefficient This parameter has the same format as theα-HTL filter coefficient and is converted to engineering units the following way:

- Ocean mode

$$\beta\_htl\_coeff = \frac{telemetred\_value * 1e5}{time\_factor\_ocean/\left[25 * (p\_ref/100)\right]} \tag{5.2.4}$$

- Ice mode

$$\beta\_htl\_coeff = \frac{telemetred\_value * 1e5}{time\_factor\_ice/\left[12.5 * (p\_ref/100)\right]} \tag{5.2.5}$$

α-STL filter coefficient This parameter has the same format as theα-HTL filter coefficient and is converted to engineering units with the expression:

$$\alpha\_stl\_coeff = \frac{telemetred\_value * 10}{1/\left(p\_ref/100\right)^2} \tag{5.2.6}$$

β-STL filter coefficient This parameter has the same format as theα-HTL filter coefficient and is converted to engineering units the following way:

$$\beta\_stl\_coeff = \frac{telemetred\_value * 100}{1/\left(p\_ref/100\right)^2} \tag{5.2.7}$$

α-AGC filter coefficient This parameter has the same format as the α-HTL filter coefficient and is converted to engineering units with the expression:

$$\alpha\_agc\_coeff = \frac{telemetred\_value}{1/\left[0.23 * \left(p\_ref/100\right)\right]} \tag{5.2.8}$$

β-AGC filter coefficient This parameter has the same format as the α-HTL filter coefficient and is converted to engineering units the following way:

$$\beta\_agc\_coeff = \frac{telemetred\_value * 1e4}{1/\left[0.23 * \left(p\_ref/100\right)\right]} \tag{5.2.9}$$

Power reference value This parameter has the following format:

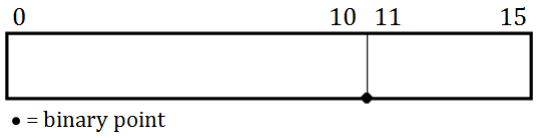


Figure 5.2.5: Power reference's format.

and shall be processed this way:

$$p\_ref = telemetred\_value * 100 \tag{5.2.10}$$

Preset duration This parameter shall be converted to integer with:

$$pre\_duration = int\,(telemetred\_value) \tag{5.2.11}$$

Preset time delay This parameter, as shown in Table 5.1.4, is telemetred in two words. The way they should be extracted depends on whether it is in ocean or ice mode.

- Ocean mode

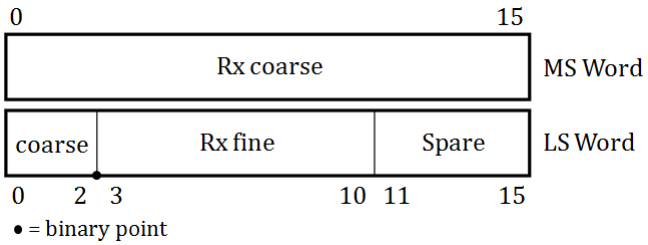


Figure 5.2.6: Preset time delay's format in ocean mode.

- Ice mode

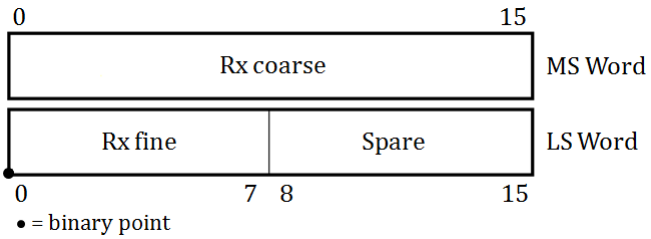


Figure 5.2.7: Preset time delay's format in ice mode.

Preset time delay rate The preset time delay rate is the first derivative of the preset time delay and its format depends on the mode too. Its format is the following:

- Ocean mode

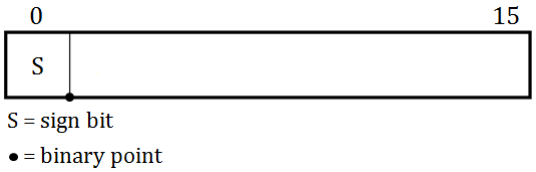


Figure 5.2.8: Preset time delay rate's format in ocean mode.

- Ice mode

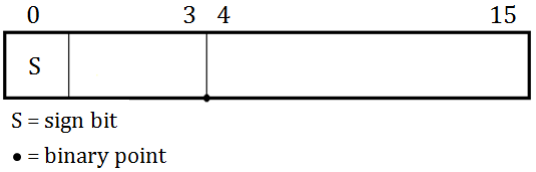


Figure 5.2.9: Preset time delay rate's format in ice mode.

Further, it has to be converted to engineering units as shown in the following equation.

$$pre\_t\_delay\_rate = telemetred\_value * 1e6 \tag{5.2.12}$$

Preset AGC This parameter has the following format:

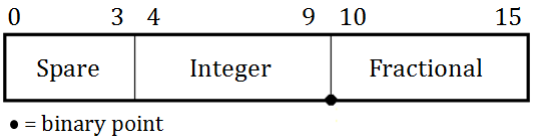


Figure 5.2.10: Preset AGC's format.

and has to be converted to engineering units with:

$$pre\_agc = telemetred\_value * 100 \tag{5.2.13}$$

Preset slope This parameter has the same format as the power reference value.

Receiver offset The last parameter of the Auxiliary Data Block has this format:

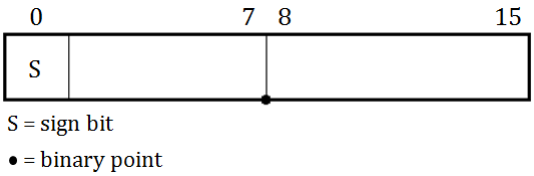


Figure 5.2.11: Rx offset's format.

Science Data Blocks

Mode identifier These two first bytes of the science data block (from now on, mode ID) contain information about the data block operating status. The possible values of the mode id are shown in the following table:

Definition	Bit	Setting
Tracking on Ocean from Acquisition	0	Off = No / On = Yes
Tracking on Ice from Acquisition	1	Off = No / On = Yes
Tracking on Ocean from Preset Tracking	2	Off = No / On = Yes
Tracking on Ice from Preset Tracking	3	Off = No / On = Yes
Tracking on Ocean from Tracking on Ice	4	Off = No / On = Yes
Tracking on Ice from Tracking on Ocean	5	Off = No / On = Yes
Closed Loop Calibration	6	Off = No / On = Yes
Open Loop Calibration	7	Off = No / On = Yes
Unused	8	-
Test / RSS mode	9	Off = No / On = Yes
Transmitted Chirp	10	Off = Ocean / On = Ice
Ground Calibration	11	Off = No / On = Yes
Unused	12	-
Unused	13	-
Ice Tracking Point	14-15	00 = n/a; 10 = 0.25 01 = 0.75; 11 = default

Table 5.2.2: Bits of mode ID.

The first 6 bits indicate whether the measurements have been taken over ice or over ocean and are activated depending on the previous instrument mode. Both Closed Loop Calibration and RSS mode are the modes used in the original calibration

concept but, after ERS-1 launch, are only used for testing purposes. Otherwise, Open Loop Calibration and Ground Calibration are the modes used to perform the calibration while the Transmitted Chirp bit is used to identify which pulse was sent (the one used over icy surfaces or over ocean, approximately 84 MHz and 340 MHz respectively). Finally, in the case of ice mode, the Ice Tracking Point is used to well decode the waveform data.

As it happened in 5.2, some variables have a different format depending on the mode (ocean or ice) so we will also make a selection, but this time based on whether bit 10 is on or not, that is, whether the ocean or the ice chirp has been sent.

Noise floor estimation This parameter has the same format as the Power reference parameter (see Figure 5.2.5) and has to be converted to engineering units the same way:

$$noise\_floor = telemetred\_value * 100 \tag{5.2.14}$$

HTL discriminator output This parameter is telemetred as a 2's complement fixed point integer value and has a different format for each mode:

- Ocean mode

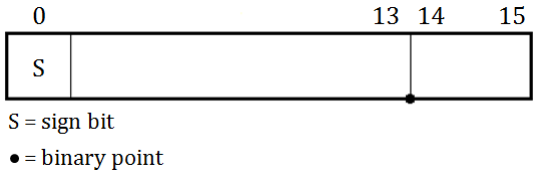


Figure 5.2.12: HTL discriminator output's format in ocean mode.

- Ice mode

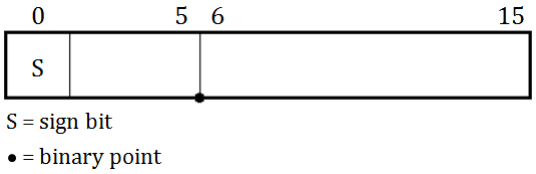


Figure 5.2.13: HTL discriminator output's format in ice mode.

STL discriminator output This parameter is telemetred as a 2's complement mantissa/exponent pair floating point as it shows the following figure:

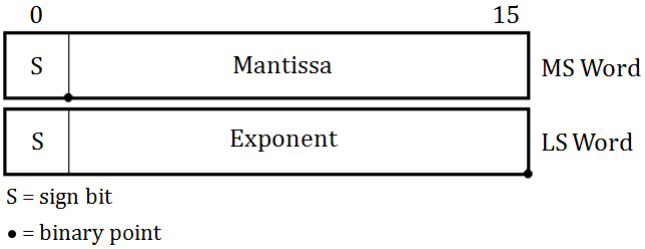


Figure 5.2.14: STL discriminator output's format.

To convert it to engineering units, the next transformation shall be applied:

$$stl\_discr\_out = \frac{telemetred\_value * 100}{(p\_ref/100)^2} \tag{5.2.15}$$

AGC discriminator output This parameter is a 2's complement fixed point integer value with the following format:

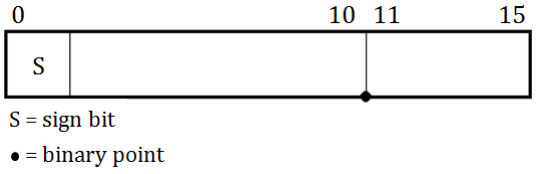


Figure 5.2.15: AGC discriminator output's format.

It shall be converted to engineering units with the following expression:

$$agc\_discr\_out = \frac{telemetred\_value * 10}{[0.23 * (p\_ref/100)]} \tag{5.2.16}$$

HTL beta branch This parameter is a 2's complement fixed point integer value and has different format in ocean mode than on ice mode.

- Ocean mode

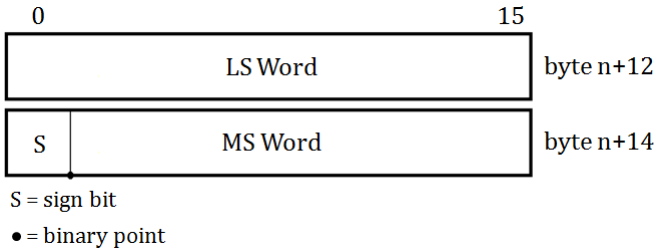


Figure 5.2.16: HTL beta branch's format in ocean mode.

- Ice mode

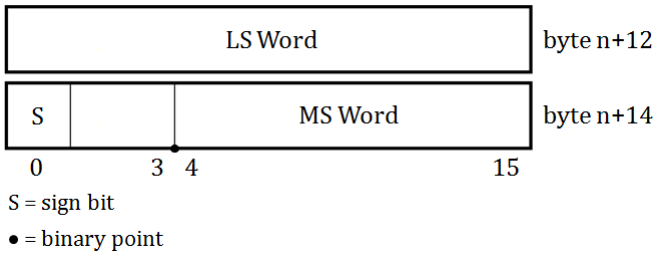


Figure 5.2.17: HTL beta branch's format in ice mode.

FFT samples This parameter contains 64 waveforms samples, which shall be treated as unsigned 16 bit integer each. Their format is the same as the power reference value. The waveform samples as directly read from the product are:

0	1	2	...	30	31	32	33	...	61	62	63

where 0 is the first sample read from the product and 63 the last one.

Table 5.2.3: Order of the samples directly read from the ERAC.

However, they do not correspond to the real position of the sample within the waveform. The real order, in which they sall be reordered, is different in each mode (ocean or ice) and are the following:

- Ocean mode

30	31	32	...	62	63	0	1	...	27	28	29

Table 5.2.4: Real order of the samples in ocean mode.

- Ice mode

32	33	34	...	62	63	0	1	...	29	30	31

Table 5.2.5: Real order of the samples in ice mode.

Time delay This parameter has the same format as the preset time delay in ice mode (see Figure 5.2.7).

Slope The parameter has the same format as the power reference value and shall be converted to engineering units with the following formula:

$$slope = telemetred\_value * 100 \tag{5.2.17}$$

AGC The parameter has the same format as the preset agc value and has to be converted to engineering units the same way.

5.3 Selection of data

In ERS, as we have seen in Section 5.2, there are several instrument modes. During the processing, we shall perform selections at two different levels: Source Packets and Data Blocks.

The selection at Source Packet level is made depending on the bits activated in the packet ID (see Table 5.2.1). In our particular case, as we are only interested in measurement packets, we will check whether bits 8 or 9 are set to 1. Even we could identify the two measurement modes (ocean and ice) with these two bits, the selection of these modes will be done at data block level. Nevertheless, there is a particular case at Source Packet level that is considered a dummy packet so it shall be discarded and not considered as a measurement packet even bits 8 or 9 are activated. This case occurs after a transition from Pause mode, indicated by all the instrument mode bits (8-15) of the packet ID set to 1.

On the other hand, the selection at Data Block level is based on the bits activated in the mode ID (see Table 5.2.2). This second selection will allow us to distinguish measurement data from OLC data and identify the Data Block's mode: ocean mode or ice mode. As we have seen in the previous section, this identification is essential to decode several variables of each Data Block.

In order to identify OLC data, as it is only contained in the first two Data Blocks of a SP, we only need to check the mode ID of these. The OLC data is present whenever bit 7 of both Data Blocks is set to On. If this condition is met, PTR data shall only be extracted from block #0 (first DB)[2]. Data from block #1 to #19 will be discarded in the OLC chain but yet stored in the measurement chain together with the first DB.

Further, once the OLC Data Blocks have been identified, another check has to be done. Even the OLC mode for tracking Source Packets shall only contain PTR data, we must assure we are only selecting PTR data and discarding all other calibration data. The method used to distinguish PTR data from other data is analysing the waveform width. PTR width should be comprised between 3 and 5 samples in ice mode and between 4 and 7 in ocean mode. Hence, other data should be larger than 5 or 7 samples. The computation of the waveform width will be done in two steps. Firstly, all the samples equal or greater than a threshold (noise floor estimation) will be evaluated and a binary vector will be created as

[2]At this point, the issue was that the old documentation stated the waveform was stored in the second Data Block.

follows:

$$\begin{cases} B_i = 1 & if\ P_i >= noise\_floor \\ B_i = 0 & if\ P_i < noise\_floor \end{cases} \tag{5.3.1}$$

where B_i is the binary value for the i-th sample, P_i is the i-th sample of the waveform and $noise\_floor$ is the threshold contained in the Data Block. Secondly, the width is computed using the following equation:

$$W = \frac{\left(\sum_{i=-N/2}^{N/2-1} B_i\right)^2}{\left(\sum_{i=-N/2}^{N/2-1} B_i\right)} = \sum_{i=-N/2}^{N/2-1} B_i \tag{5.3.2}$$

where W is the width and N is the number of waveform samples (64).

However, as it happened at Source Packet level, there are dummy Data Blocks and they must be identified. The following cases are considered dummies:

1. An anomaly in the data occurs in the first tracking SP of a set after acquisition, preset tracking or BITE mode. In this case, the first Data Block does not contain valid FFT samples, neither any quantity derived from it.

2. The last science data block before acquisition.

3. A science data block with all the bits from its mode ID set to 1.

4. The first Data Block of an ERAC.

To sum up, and in order to clarify as much as possible the whole process of data selection, the figure below shows the steps taken at each level.

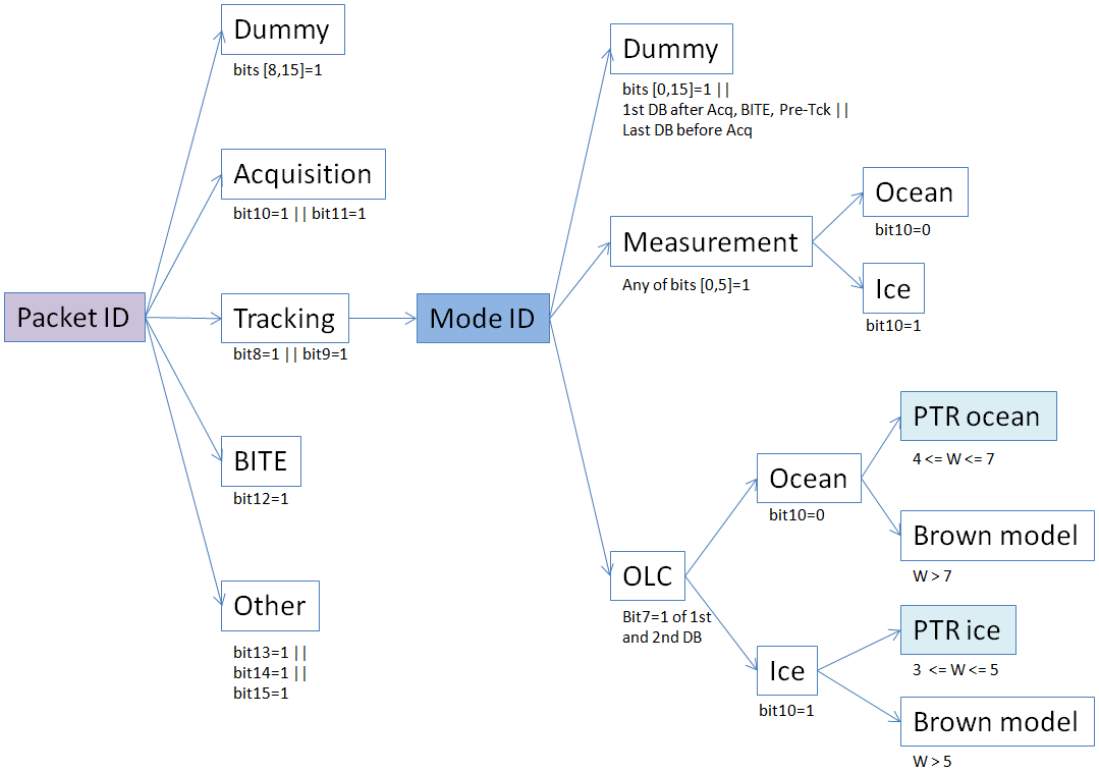


Figure 5.3.1: Steps to select all the data.

Bibliography

[1] http://earthobservatory.nasa.gov/Glossary/

[2] http://www.wmo.int/pages/prog/wcp/ccl/faqs.php

[3] Solomon, S., D. Qin, M. Manning, Z. Chen, M. Marquis, K.B. Averyt, M. Tignor and H.L. Miller (eds.). *Climate Change 2007 - The Physical Science Basis. Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change IPCC*. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA (ISBN 978 0521 88009-1 Hardback; 978 0521 70596-7 Paperback).

[4] UNFCCC (1992). *United Nations Framework Convention on Climate Change*. FCCC/INFORMAL/84, GE.05-62220 (E) 200705.

[5] http://cci.esa.int/

[6] https://earth.esa.int/web/sppa/mission-performance/esa-special-series/reprocessing-of-altimeter-products-for-ers

[7] http://www.altimetry.info/radar-altimetry-tutorial/

[8] C. Elachi. *Spaceborne radar remote sensing: applications and techniques.* IEEE Press, New York, 1988.

[9] I.G. Cumming, F.H. Wong. *Digital processing of synthetic aperture radar data: algorithms and implementation.* Artech House, Norwood, MA. January 2005.

[10] M. Roca. *Comprehensive Survey of Past, Present and Near Future Radar Altimeter Systems*. ESTEC, ESA, Noordwijk, March 1996.

[11] M. Roca, S. Laxon and C. Zelli. *The EnviSat RA-2 Instrument Design and Tracking Performance*, in IEEE Transactions on Geoscience and Remote Sensing, vol. 47, no. 10, pp. 3489-3506, Oct. 2009. doi: 10.1109/TGRS.2009.2020793

[12] https://earth.esa.int/web/guest/missions/esa-operational-eo-missions/ers/satellite

[13] C.R. Francis, M. Roca. *SPTR Algorithm Specification*. ER-TN-ESA-RA-0150, v1. ESTEC, ESA, Noordwijk, 28 October 1998.

[14] ESA. *ERS-1 Satellite to Ground Segment Interface Specification*. ER-IS-ESA-GS-0001, v2.c . February 1991.

[15] isardSAT, for ESA (internal document). *RA Level 1b DPM*. REA-DD-DPM-L1b-6001, v4e. ESA. 8 April 2014.

[16] isardSAT, for ESA (internal document). *RA L1b IODD*. REA-DD-IODD-L1b-6002, v4c. ESA. 8 April 2014.

[17] Altimetric LLC, for ESA. *RA L2 Validation Report*. REA-DD-VAL-L2-7001, v3.1. 23 July 2014. Available at [6].

[18] CLS, isardSAT. *WP2100: Altimeter Instrumental Processing: RRDP and Validation Reports*. Barcelona. May 2012.

[19] Ablain, et al. (2015). *Improved sea level record over the satellite altimetry era (1993-2010) from the climate change initiative project*. Ocean Sci. 11, 67–82. doi: 10.5194/os-11-67-2015